DOCUMENT RESUME

ED 114 110 IR 002 740

AUTHOR Richards, William D., Jr.
TITLE A Manual for Network Analysis (Using the NEGOPY Network Analysis Program).
INSTITUTION Stanford Univ., Calif. Inst. for Communication Research.
PUB DATE Jun 75
NOTE 94p.

EDRS PRICE MF-$0.76 HC-$4.43 Plus Postage
DESCRIPTORS Algorithms; Computer Programs; *Information Theory; *Intercommunication; Models; *Networks; *Social Systems; *Topology
IDENTIFIERS NEGOPY; *Network Analysis Program

ABSTRACT

Network Analysis is an observational system focusing on the relationships of individuals within a system--their subgroups, their leaders, and the frequency of their communications. The strength of relationships can be measured by examining the incidence and duration of intercommunications. Using data accumulated by questioning each member of the organization, the computer employs a prescribed set of algorithms, the Network Analysis Program (NEGOPY), to yield a topological interpretation of individuals, subgroups, and inte communications. (EMH)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED114110

a report of the

INSTITUTE FOR COMMUNICATION RESEARCH
STANFORD UNIVERSITY

IR 002 740

ED114110

A MANUAL FOR NETWORK ANALYSIS

(USING THE NEGOPY NETWORK ANALYSIS PROGRAM)

William D. Richards, Jr.
Institute for Communication Research
Stanford University

June 1975

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY
William D. Richards, Jr.
TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER.

© Copyright 1975, William D. Richards, Jr. No part of this document, with the exception of short quotations, may be copied or reproduced in any form without the prior written consent of the author.

TABLE OF CONTENTS

Part one--The goals of Network Analysis.....2

Part two--The data for Network Analysis.....8

Relationships.....8

Ratio level scales.....11

Combining two or more indicators.....12

Other instrumentation considerations.....15

Directionality/symmetry.....18

The actual collection of data.....18

Part three--The analysis.....21

I. The algorithm which defines groups.....22

Drawing the tentative boundaries.....30

Using the criteria for an exact solution.....38

II. NEGOPY: the Network Analysis Program.....46

1. Data description

a. review.....48

b. parameters.....48

c. output.....55

2. Preparation for group detection

a. review.....59

b. parameters.....60

c. output.....61

3. Initial group detection

a. review.....61

b. parameters.....61

c. output.....63

4. Application of the formal criteria

a. review.....64

b. parameters.....65

c. output.....66

5. Final results and control of output

a. review.....68

b. parameters.....68

c. output.....70

table of contents, continued

Part four--Using the Network Analysis Program.....75
I. Setting up a Network Analysis run.....75
System control cards.....76
NEGOPY control cards.....77
Data cards and data format cards.....79
Namelist cards and namelist format cards.....81
II. Miscellaneous aspects of running the program
Error messages or warnings.....82
Adjusting the parameters to get better results.....83
Known bugs.....84

Parameter List.....86

PART ONE

THE GOALS OF NETWORK ANALYSIS

Without communication, there would be no social organizations. There would be no corporations, no hospitals, no universities, no societies. The process of communication allows people to work together for some common goal. It allows people to coordinate their behaviors and to share their feelings. Communication has been described as the "thread" that holds organizations together, as the "glue" that bonds people together in relationships, and as the force that allows groups of people to take on their own identity. Clearly, the process of communication is fundamentally important to any activity that requires more than a single person.

As the number of people working together toward a common goal increases, as the complexity of that goal increases, so does the importance of communication increase: the more people there are, the more important it becomes to keep everyone informed and the more important it becomes to efficiently coordinate the behaviors of the people. The more complex the task of the organization, the more important that everyone know their job and the more important an efficient information flow becomes.

Network Analysis is a way of studying the communication networks that develop in social systems as people communicate with each other. Specifically, it is a way of examining the whole set of relationships that exists in a functioning, ongoing system. Network Analysis allows us to make statements about intact systems because it takes a systems approach

to the analysis situation: it focuses on the relationships between the people in the system and looks at all the relationships at once, without isolating the people from each other or from the relationships between each other. With Network Analysis we can not only see how the system as a whole is structured but we can also see how each individual person fits in with the larger structure. This becomes crucial when very large systems are involved, as people may come to play specialized roles in the communication networks of these systems, and as poorly organized networks may lead to very serious problems for the organization as a whole.

Network Analysis allows us to study the system as it is, rather than as someone thinks it ought to be. This is because it uses data collected from people in the system which describes the way they fit into the system as the system normally functions, rather than relying on organizational charts which tell how people ought to behave or how management thinks people behave. Thus, the information provided by Network Analysis is more valid than other kinds of information.

With Network Analysis, we can study the structure of the system. Large systems are almost always differentiated into smaller parts. These smaller parts may be groups of individuals who work together on a common task or they may be groups of groups of individuals. Network Analysis identifies these groups and shows us how they are connected, either by direct links between members of the different groups or by links that go through specialized "linkers" -- people who function as "go-betweens" or "liaisons" to connect the groups. Network Analysis also gives detailed descriptions of the communication flows within the groups as well as flows between groups. For example, we can see if some people in the group are

more central or more critical, in terms of information flows within the group, than others in the group.

Since Network Analysis is based on networks -- sets of nodes (people) with links (communication relationships) between them -- it gives us topological information. This kind of information may be contrasted with distance information which is provided by multidimensional scaling methods or with variance information which is provided by correlational methods. Since we are interested in topological properties of networks -- who talks to who, and so on -- we should use a method that takes this approach. To be sure, we may later be interested in other kinds of information, and then we would use other techniques. The Network Analysis method described here is primarily topological.

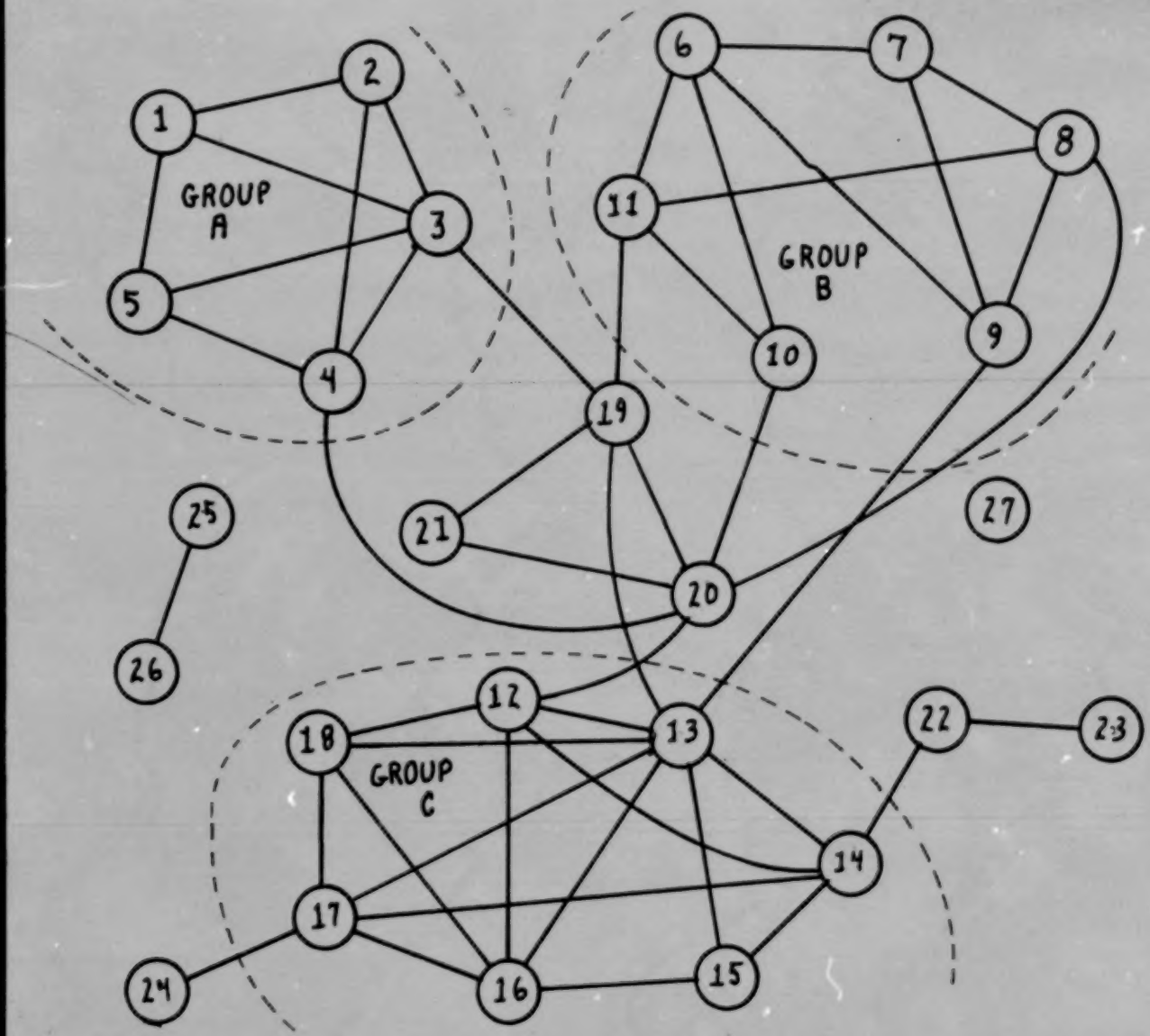
Many other researchers have attempted to do Network Analysis in the past. These investigators have used a variety of analytic techniques, including sociograms, matrix multiplication or manipulation methods, and factor analysis. None of these methods are ideal: all are slow and cumbersome, some do not work at all, and the others do not work well. The common failure of these other approaches can be traced, among other things, to a lack of clear definitions and unambiguous goals. Indeed, this is the point at which we begin -- with a clear set of goals and formal definitions. We use concepts that appear to be very similar to the concepts used by the early sociologists -- groups, liaisons, and so on. However, our definitions are explicit and clear, and this will allow us to do much more than would be possible otherwise. The definitions we use are as follows.

We begin with the smallest units of analysis -- <u>nodes</u> and <u>links</u>. In the case where we are examining a communication network in an organization,

nodes would be people. Whenever a person reports a relationships with another person, there would be a link between the corresponding pair of nodes.

We then divide up the nodes into two types -- participants and non-participants. Participants are the nodes that take part in the exchange and transfer of information with other participants. The non-participants include all the nodes having either no connection or only minimal connection to participants. There are four kinds of non-participants (all the roles are illustrated in Figure 1).

Isolate Type 1	These nodes have no links whatsoever.
Isolate Type 2 or Attached Isolate	These nodes have only a single link and thus cannot take part in the transfer of information through the network. They may, however, function as sources of information if they have links outside the system.
Isolated Dyad	These nodes are similar to attached isolate pairs who are linked to each other. In terms of contact with the rest of the network, they function more like Isolates Type 1.
Tree Node	If there is a subset of nodes with minimal connections (i.e., the number of links, , equals n-1, where n is the number of nodes in the subset), the subset will be a tree structure composed of isolates and tree nodes. The removal of any single links separates the tree structure into two isolated parts. The isolates will be the nodes at the ends of the structure, and will have only single links. The tree nodes are the other nodes in the structure.



GROUP MEMBERS	LIAISONS	OTHERS	ISOLATE-1	ISOLATE-2	TREE NODE
GROUP A 1,2,3,4,5	19, 20	21	27	24, 23	22

GROUP B

6, 7, 8, 9, 10, 11

GROUP C

12, 13, 14, 15, 16, 17, 18

ISOLATED DYAD

25, 26

Figure 1 – NETWORK ROLES

Participants are nodes that have two or more links to other participant nodes. They make up the bulk of the network in most cases and allow for the development of structure. They include:

A. Group member. A node with more than some percentage of his linkage with other members of the same group. (This percent is called the alpha-percent or α-percent.)

B. Liaison. These nodes fail to meet the α-percent criterion with members of any single group but do meet it for members of groups in general.

C. Type other. These nodes fail to meet the α-percent criterion for any set of group members.

To be called a group, a set of nodes must satisfy these five criteria.

A. There must be at least three members.

B. Each must meet the α-percent criterion with the other members of this group.

C. There must be some path, lying entirely within the group, from each member to each other members. (This is called the connectiveness criterion.)

D. There may be no single node (or arbitrarily small set of nodes) which, when removed from the group, cause the rest of the group to fail to meet any of the above criteria. (This is called the critical node criterion.)

E. There must be no single link (or subset of links) which, if cut, causes the group to fail to meet any of the above criteria. (This is called the critical link criterion.)

The following points are relevant here. First, the "α" that appears above is usually set to something greater than 50%, such as 51% or 50.01%. This is done to prevent ambiguous situations which could occur otherwise.

Second, several of the criteria refer to a proportion of a node's linkage. We are referring here to amount of linkage rather than number of links. If amount is operationalized as "time spent interacting," we would look at the appropriate sets of links in terms of what fraction of the total amount of time they comprise, rather than in terms of how many links total there are. Thus, a node having ten links could be a group member even if only 2 out of those ten links (20%) were with members of the group, as long as those two links account for more than 51% of the total linkage. For example, those two links might take four hours per week. If the eight other links combined total less than four hours per week, the node would be a member of the group.

The goals of Network Analysis are to classify the nodes in the network into the various roles, based on their patterns of interaction with each other and to provide as much information as possible about the system at each of three levels -- the individual, the group, and the whole system.

PART TWO

THE DATA FOR NETWORK ANALYSIS

We said earlier that this approach is different from others because it focuses on the relationships between individuals rather than on the individuals themselves. Indeed, it is this focus on the relationships that allows us to make statements about the system as a whole rather than merely about the people in the system. Because the relationships play such a central role in the conceptual formulation, the way we handle them deserves a very careful examination. This is the area we cover in this section.

Relationships

We might begin by asking which aspects of relationships are important to us. There seems to be an almost infinite variety of kinds of relationships between persons in organizations or systems. If we are to make any progress at all, we must reduce this infinite variety to a smaller, more useful set of dimensions. Of course, this has been done. For our purposes, we only need to look at a few aspects of relationships between persons. One is the strength of the relationship. How much does it matter? How often is the relationship "activated" or "used"? Another is symmetricality -- if I am related to you, are you related to me in the same way? Another is transitivity -- if I am related to you and you are related to Harry, does that mean I am related to Harry?

If we are working with communication networks, the relevance of these questions becomes clear. The strength of the relationship might refer to how often we talk, or to how much information we exchange. The relationship might be symmetrical, where we share ideas and exchange information, where the influence is bi-directional; or assymetrical, where I would give you information or you would give me orders, where the influence goes in one direction only. Similarly, if I pass on information I receive from you to others, the relationship would be transitive, while if I do not, if I keep it to myself, the relationship would be intransitive.

Surely there are other factors that are important. Perhaps the most important, at least in the case of a communication relationship, is the content or function of the relationship. In this way we might distinguish between formal, job-related communication and informal communication about matters not related to the job, or between communication about new ideas and communication about errors or problems that are encountered in day-to-day work.

These are the factors that are usually considered to be important by people who do a lot of network analysis. But these are all conceptual factors. They all refer to the relationship between the people in the system. These conceptual issues must be translated into operational procedures so that we can build a model of the system in the form of data. This translation is accomplished by our operationalization of concepts and by our measurement procedures. Here we will create artificial constructs which we will use to represent the real world.

For example, in Network Analysis we refer to people as nodes. The node is our representation of the person. Similarly, we use links to

represent relationships. The link is not the relationship; it indicates that there is a relationship between the people corresponding to the nodes it connects. This may seem to be a useless philosophical complication, but it is important to keep the distinction clear. Let us see why.

If we conceptualize the relation as symmetrical, then whenever A is related to B, B should also be related to A. This means, for example, that if we are using the relationship "talks to" as symmetrical, if Harry says he talks to Joe, Joe should also say he talks to Harry. But this might not be the case. If Joe does not say he talks to Harry, one of two things might be happening. First, the relationship might not be symmetrical. Maybe Joe really does not talk to Harry. Maybe Harry talks at Joe. Second, perhaps Joe simply forgot that he talked to Harry. Maybe he just made a mistake and the conceptualization of the relation as symmetrical was accurate.

Thus, we can use the correspondence between relationships and links to verify our conceptualization and to check on our measurement techniques. If a relationship is conceptualized as symmetric, all links should be reciprocated. If this does not happen, we have to decide either to alter our conceptualization or place less confidence in our measurement technique. If only a few links are not reciprocated, we might assume the problem is measurement error and either delete unreciprocated links or add the "missing halves." If a large percentage of our links are unreciprocated, we should consider the possibility that either the relationship is really not symmetrical or that there is a serious measurement problem which is biasing our results in a direction that makes things look as if they do not fit our conceptualization. In other words, there

may be a symmetrical relationship but our measurement process may be getting at a separate, asymmetrical relationship.

A second set of considerations is related to the strength of the relationship. In order to use the conceptual system we discussed earlier as a classification scheme for nodes, we have to have some indicator of the strength of the relationships between them. When we are interested in communication networks, a logical indicator of strength is "the amount of information exchanged or passed on from one node to the other." Now, this is a difficult quantity to measure. We might assume, for the sake of simplicity, that the amount of information flowing is proportional to the length of time spent communicating, perhaps, or to the frequency of interaction. Our actual measurement procedure would then tap the duration or frequency of interaction. In this case, we would ask the people in the organization to indicate who they talk with and either how often or for how long.

Ratio Level Scales

Not only does this indicator of the strength of the relationship have to be a single number but it must also be a ratio-level indicator. That is, it must vary approximately as a ratio of the strength of the relationship. This is easily accomplished with the appropriate choice of coding systems. If we ask people how much time in minutes they spend talking to the people they talk with, we will have a ratio-level indicator. If we provide categories like:

Once a month or less
Once or twice a week
Once or twice a day
Several times a day

we will have to assign numbers to these categories in such a way that a ratio-level approximation is achieved. If we translate the categories into number of interactions per month, we might get:

CATEGORY	CODING
Once a month or less	= 1
Once or twice a week	= 8
Once or twice a day	= 27
Several times a day	= 64

If we code a "once a month" response as "1", a "once or twice a week" as "8", and so on, we will have an approximation to a ratio scale. Rather than using the numbers "1, 8, 27, 64," we might use their cube roots (which happen to be 1, 2, 3, 4) and restore the original values at the time of analysis.

Combining Two or More Indicators

We may decide that simple frequency or duration data are not good enough. For example, a very important exchange might be very infrequent or short, and we might want this to balance with exchanges that are individually less important but occur much more often. In this case, we might ask our respondents to indicate how important the information exchanged was, in addition to how much or how often. We would then combine the two scales into a single indicator of the strength of the relationship. Let us take an example to show how this is done. Say we ask both frequency and importance questions, as shown below.

Please indicate by circling the appropriate numbers which people you talk to, how often you talk to them, and how important the interaction usually is. Use the coding system shown here.

FREQUENCY	IMPORTANCE
1 = once/month	1 = slightly important
2 = once/week	2 = moderately important
3 = once/day	3 = very important
4 = several times/day	4 = crucial to survival

NAME	FREQUENCY	IMPORTANCE
John Jones	1 2 3 4	1 2 3 4
Emily Stuart	1 2 3 4	1 2 3 4
Tony Mann	1 2 3 4	1 2 3 4
Belinda Humm	1 2 3 4	1 2 3 4
Mark Smith	1 2 3 4	1 2 3 4

We would form a matrix where the rows are for the different values of frequency and the columns are for the different values of importance, as shown here.

FREQUENCY \ IMPORTANCE	Slightly	Moderately	Very	Crucial
Several/day	?	?	?	?
Once/day	?	?	?	?
Once/week	?	?	?	?
Once/month	?	?	?	?

We would then decide which entries should have the highest and lowest values. Obviously, these would be the top right entry and the bottom left one in the example.

The next step is to assign the intermediate values. This is more difficult. For example, how does the top left entry compare with the bottom right one? What about other entries? If the values shown below are acceptable, the two scales can simply be multiplied together to give the final results. If importance had been coded in the reverse order, as shown below, the values for that

scale would have to be reversed. In this case, they would be subtracted from five to give the results shown here.

4	4	8	12	16
3	3	6	9	12
2	2	4	6	8
1	1	2	3	4
	1	2	3	4

The Matrix With Values Filled In

ORIGINAL CODE	IMPORTANCE	REVERSED CODE
1	Crucial	4
2	Very	3
3	Moderately	2
4	Slightly	1

Reversing Scales to Obtain the Correct Orders

In the example discussed above, we formed the strength indicator by taking the cross product of the two original scales. In other cases, we would use a linear combination instead. For example, say we had separate scales for face-to-face and telephone interactions, as shown here.

Please indicate how much time you spend talking to each person in an average week (in minutes)

NAME	FACE-TO-FACE	TELEPHONE
Robert		
James		
Annie		
Frank		
Susan		

We might decide that face-to-face interactions are twice as important a telephone interactions because of the additional non-verbal information that is transmitted in face-to-face interactions. Then we would use this formula for calculating the final strength indicator:

Strength = 2*Face-to-Face + Telephone

The important point here is that a single ratio-level indicator (or an approximation to one) must be available as an index of the strength of the relationship. A lot of trouble can be saved by constructing instruments so that they can be easily coded to give ratio-level data. If this is not done, the data must be transformed to give ratio data at the time of analysis, if that is possible.

Other Instrumentation Considerations

In the discussions above we have seen several examples of instruments that might be used to collect network data. They are all variations of the same basic design. Some types seem to work better than others in different situations. For example, there are two ways of getting the respondent to provide the names of the people he or she is linked to. The first works well when there are less than about two or three hundred people in the organization. With this method, a list of all the people is provided and the respondent simply fills in the appropriate spots on the instrument. An example of this type is shown in "A" below.

How often do you interact with the people named here? Please indicate the approximate number of interactions per week for both job-related conversations and other conversations.

NAME	JOB-RELATED	OTHER
Sam		
Mary		
Bill		

A

In the column on the left, please write the names of people you talk to. In the other columns please indicate how many times you talk to these people in a typical week. Do this for both job-related conversations and other conversations.

NAME	JOB-RELATED	OTHER

B

In the first type, the respondent only has to recognize the name of the person he or she is linked to. In the second type as shown in B, the respondent is asked to recall the names. The second type is appropriate for very large organizations, where it would be impractical to provide a list of all the names because of the length of such a list, or for systems where all the names of relevant people are not known.

There is likely to be a difference in the number of contacts reported on the two types of instruments. Specifically, since it is easier to recognize a name on a list than to recall a name from memory because the list of names serves as a prompter, there will probably be more contacts reported with the first method than with the second. Unfortunately, this difference has not been tested empirically so no definite statements can be made regarding the trade-off.

A second way in which instruments may vary is in the method of coding the strength of interactions. A variety of approaches has been used here: (a) interaction frequency may be coded into categories as shown in "A" below; (b) interaction frequencies may be coded directly, as shown in "B"; (c) interaction duration may be coded into categories, as shown in "C"; (d) interaction duration may be coded directly, as shown in "D".

FREQUENCY		DURATION	
1. Once/month 2. Once/week 3. Once/day 4. Several/day	How many times in the last week?	1. Less than 5 mins. 2. Less than 10 mins. 3. Less than 20 mins. 4. Less than 30 mins. 5. More than 30 mins.	How much time in the last week?
A	B	C	D

From a theoretical perspective, it would seem that the method shown in "D" above would provide the most valid information. However, it is harder to estimate durations of interactions than frequencies of interaction, as in "A" and "B", and it is harder to estimate precise numbers than simple categories, as in "A" and "C". Thus, the method shown in "A" is probably the easiest for subjects to use, while the one in "D" provides the best information. Again, there have been no empirical studies comparing the alternative methods.

When several content areas are to be used at once, it is not necessary to have a separate instrument for each one. Instead, they can be combined into a single form, with multiple columns for the different content areas. An example of this is shown below, where three separate content areas are being measured at once. In analysis, these will be treated as three separate networks which might later be compared and examined for similarities or differences.

Please indicate how often you talk to the following people about each of the three topic areas. Use this system for coding your responses.

1. Once a month
2. Once a week
3. Once a day
4. Several times a day

NAME	PRODUCTION: GETTING MY JOB DONE, DAY-TO-DAY MATTERS	INNOVATION: NEW IDEAS OR WAYS OF DOING THINGS	SOCIAL RELATIONS: INFORMAL FRIENDSHIP CONVERSATIONS, ETC.
Harry			
Timothy			
Maude			
Jenny			
Donald			
Michael			

Directionality/Symmetry

We mentioned symmetricality as one dimension among which relationships may vary. In order to be consistent, our instruments should reflect our conceptualization of the relationship we are interested in. In this case, we should ask a question that elicits non-directional responses if we have a symmetrical relationship, and so on. This is especially important when we are interested in asymmetrical relationships, for which we would expect the direction of the relationship to be important. For example, if we are interested in information flows, specifically in the direction of flow, we might use a question like: "Please indicate which of the following people you received information from in the last week" or "Please indicate how often each of the following people come to you for information." On the other hand, when we do not want directional information, we should be careful not to use instruments which elicit this information, since the result will be a distorted version of the network we are really interested in.

The Actual Collection of Data

Compared to face-to-face interviewing methods, the collection of network data is relatively fast and easy. Respondents can be assembled in large groups by the team of investigators who then explain the nature of the study, insure confidentiality, and discuss the instrument. Once the instrument is distributed, there will be questions about the meaning of content categories, what to do if a person's name does not appear on the list, and so on. At any rate, the time required to fill in the instrument will seldom be more than an hour.

There are several very important points to be made here. First, it is absolutely essential that respondents indicate who they are. This can be easily done by asking them to circle their own name on the list or to write it at the top of the page. If this is not done, the data are useless.

Second, for a valid Network Analysis to be done, the entire system must be censused. Every person in the system should fill out an instrument. A failure rate of more than five to ten percent will greatly reduce the validity of this technique. Thus, if some respondents miss the data collection session or neglect to turn in the completed instruments, it is necessary to have a member of the investigating team locate those individuals and obtain the data if possible. If this is not possible, a list of missing persons should be compiled and used in the interpretation of the final results. If the system as a whole is too large but may be broken down into smaller divisions or subsystems, a sample of these subsystems may be analyzed, where there is complete censusing within each intact subsystem. This situation is due to the nature of Network Analysis: the unit of analysis is the system, rather than the individual.

Third, respondents should be encouraged to indicate how they really behave, rather than how they think they ought to behave. It helps if the study is introduced as a diagnostic aid "to see that people get the information they need to do their jobs," and if the confidentiality of the study is emphasized: "No one from the _____ Company will see these forms. We (the Analysis team) will take them back with us to our university, where we personally will do the analysis."

Fourth, respondents must be given unique subject numbers, running from 1 to N, where N is the number of respondents in the system. If these

numbers are printed on the data collection instruments next to the names of respondents, coding and punching is greatly facilitated. This is not possible when open-ended instruments (wherethe respondent must recall the names of individuals he or she talks with) are used. In these situations, the respondent numbers must be obtained from a list of names and added at the time of coding. There are serious difficulties with this method, as people cannot recall full names of individuals they talk with, or as they write names incorrectly or illegibly. For these reasons, the other format for instruments is highly recommended.

PART THREE

THE ANALYSIS

Introduction

So far we have seen what Network Analysis does and what kind of data it uses. In this section, we will see how the actual analysis is accomplished. The main tool we have is Negopy, the Network Analysis Program.* In the first part of this section, we will discuss the algorithm upon which the program is based, because an understanding of the kinds of things the program does is useful to any potential user of the program. In the second part, we will take up several considerations directly related to the 1975 CDC version of the program, such as limits on the data, specific requirements, and so on. We will also discuss the various options the user has when running the program and the output of the program -- what the various tables mean and how to interpret them. The third part contains detailed information on using the program -- how to prepare control cards, and so on. In Part Four, several miscellaneous issues are discussed. These include error messages and how to interpret and correct them, how to "fine tune" the program, and a section on known "bugs" in the program.

The point was made earlier that Network Analysis is a topological method -- it looks for specific patterns in the data. The realization that this is a pattern recognition problem made it possible to program a computer to do the analysis. The Network Analysis Program, then, is based

* © Copyright 1975, WDR

on a pattern recognition algorithm. Although it uses a variety of statistical and mathematical operations as it carries out an analysis, it is not based on mathematical or statistical procedures, as are other kinds of analytic programs.

There are five stages in the analysis. In the first stage, the data are read in and cleaned and organized in an orderly fashion. Then an iterative operation that makes the actual pattern recognition part possible is performed in the second stage. In the third stage, the pattern recognition algorithm is carried out. Here, groups are tentatively identified. In the fourth stage the strict criteria for the various role definitions are applied and the tentative solution produced earlier is tested and made exact. The results of the analysis are printed out in the form of various tables and charts in the fifth and final stage. In the present part of the discussion, we will only cover the second, third, and fourth stages of the analysis -- the parts where groups are identified and the formal criteria are applied.

I. THE ALGORITHM WHICH IDENTIFIES GROUPS

The major task to be accomplished in this part of the analysis is to identify the groups. We have data describing the relationships between individuals. If we can represent the data in the right way, it will be easy to see the groups. The representation we would like would be one in which the members of each group are close to other members of the same group and far from the members of other groups. Then we could just look for clusters -- groups -- sets of nodes having most of their linkage to other nodes in the same groups. This will be a graphical representation

of the data -- nodes will be moved around until their locations, relative to other nodes, can be used to decide the way they fit into the network.

The first step is thus to rearrange the data so that the groups become visible, and the second to identify the groups. The way the first step is accomplished can be understood easily with the following analogy. Imagine the nodes to be like billiard balls scattered about in space. Imagine there to be rubber bands connecting the balls corresponding to nodes with links between them. Imagine there to be springs between balls corresponding to nodes that do not have links between them. The rubber bands will act to pull the balls connected to each other closer to each other, while the springs will push the balls not connected to each other apart from each other. If we hook up the rubber bands and springs and release the balls, they will rearrange themselves so that the balls corresponding to nodes with links to each other will be close to each other, while the balls corresponding to nodes that are not linked to each other will be pushed away from each other. This example is shown in Figure 3.

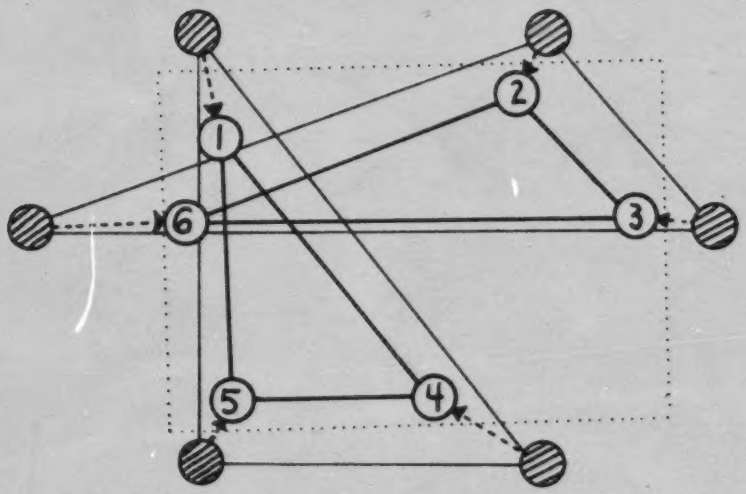
We could refine this technique by using heavier rubber bands to represent the links that occur more often or are more important. Since our objective here is to make it easier to identify groups, we could make the process work even better if we could make the rubber bands for within-group links heavier than the ones for other kinds of links. In order to do this, we will need some indicator that tells us which links look like within-group links.

If two nodes are in the same group, they are likely to have many links to the same people. There is likely to be a high number of shared links, or two-step links between this pair of nodes. If they are not in

FIGURE 3

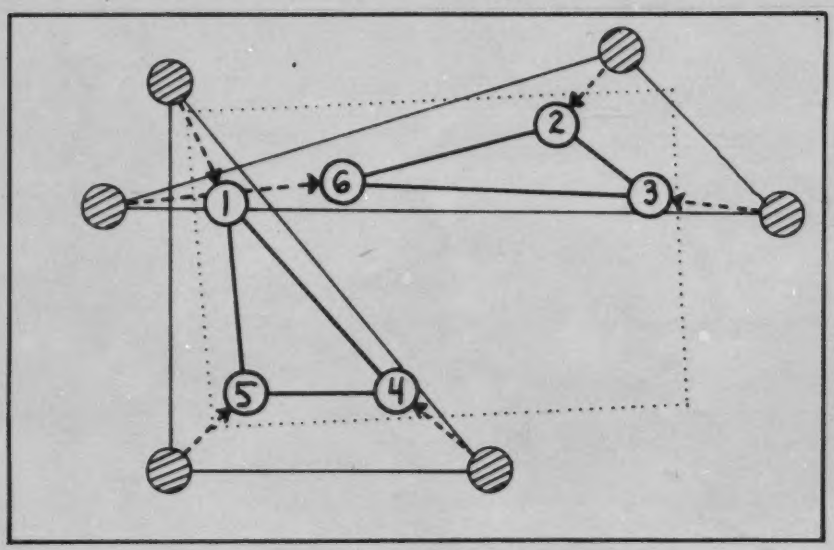
This figure illustrates the billard ball and rubber band model described in the text. The network shown has two groups of three nodes each. The three drawings represent three successive increments of time, as the nodes move farther and farther in response to the forces exerted by the rubber bands.

The original position of the balls is shown by the shaded circles in the top drawing. Movement of balls during each time increment is shown by the dotted arrows in the three drawings. The scale was changed in going from the first to the second to the third drawing in order to show smaller and smaller regions in space as occupying the same sized area in the drawings. The region of the top drawing shown in the middle one is indicated by the dotted box in the top. Similarly, the area of the bottom drawing is shown by the dotted box in the middle one.



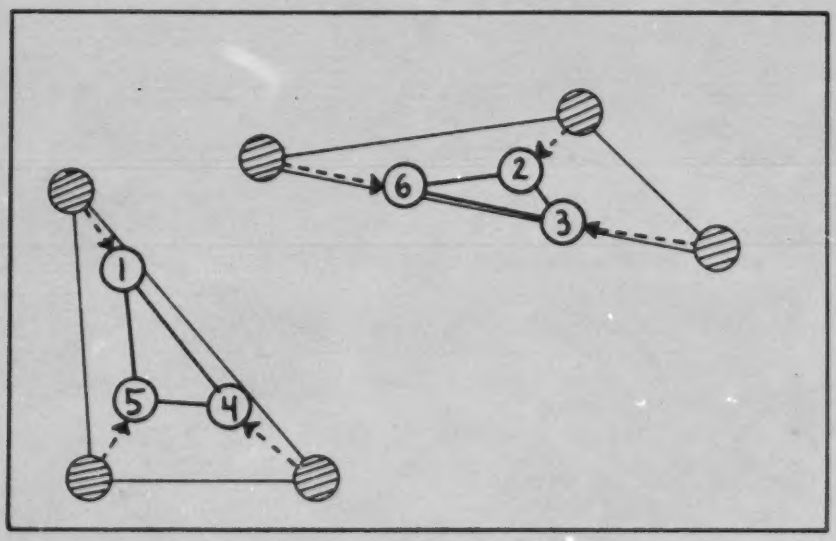
SHADED CIRCLES - NODES AT TIME 0

NUMBERED CIRCLES - NODES AT TIME 1



SHADED CIRCLES - TIME 1

NUMBERED CIRCLES - TIME 2



SHADED CIRCLES - TIME 2

NUMBERED CIRCLES - TIME 3

FIGURE 3

the same group, they are not likely to talk to the same people, and there are not likely to be many two-step links between the nodes. Thus, the number of two-step links is used as an indicator of the probability that the link is a within-group link.

Now, it is hard to represent large numbers of points in multi-dimensional space. It takes a lot of information to do this, and it is fairly difficult to move objects in this kind of a space. Extensive experimentation with real data, however, showed that it was not necessary to use a multi-dimensional representation for this analysis; a single line segment was sufficient. This kind of reduction in complexity of representation greatly reduced the amount of information needed to perform the analysis at the same time it made the analysis itself easier to do.

The analysis is performed as follows: nodes are scattered at unit points along a line segment N units long, where N is the number of nodes. We then treat each link from, say, node A to node B, as a vector, starting at A and pointing at B. We take all the vectors for each person and compute the average, weighting the individual vectors for strength of the link and probability that the link is a within-group link. We then get a single point for each individual, that point being the mean of that person's vectors. This is illustrated in Figure 4. After all the means have been computed, each node is moved to the point indicated by his or her mean.

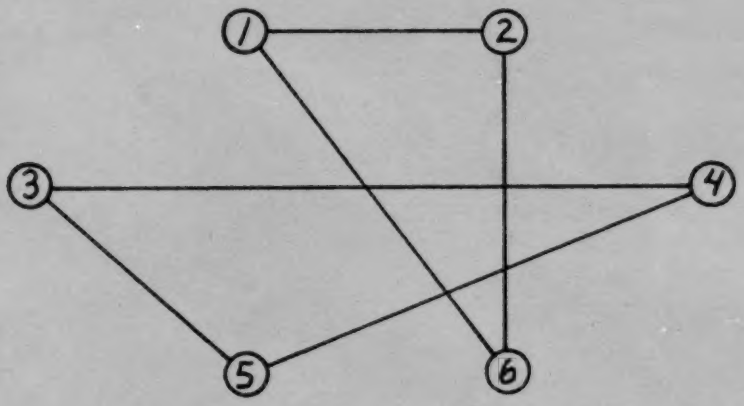
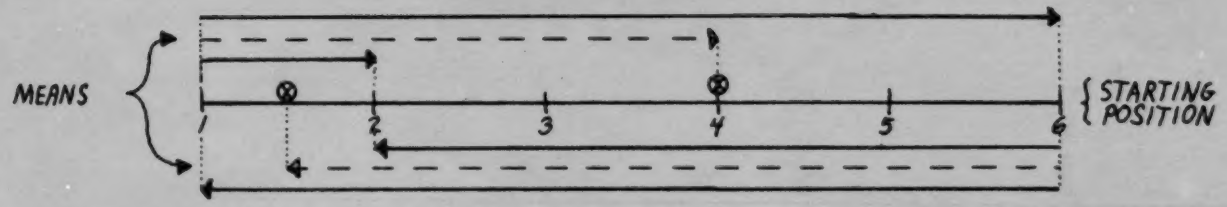
After this process has been completed, nodes with links to each other will be closer to each other than they were before. They will not, however, be as close as they could be. This fact is due to the way nodes are scattered initially, and also because of the statistical properties

FIGURE 4

At the top of this figure is shown a hypothetical network consisting of two groups, each of which has three members.

The diagram in the middle shows how the six nodes are initially placed along a line segment. The two solid arrows pointing to the right in the top of this figure are the vectors representing the links of Node #1 to Node #2 and Node #6. The dashed arrow between the solid ones is the average of the two. Below the line segment are shown the vectors for the links of Node #6.

The diagram on the bottom of Figure 4 shows how the iterative process of vector averaging works. The first line shows the initial positions of the six nodes. The second shows what the means could look like. Moving from the second to the third lines, the scale has been expanded so that the nodes range over the entire length of the continuum. The fourth and sixth lines show the second and third sets of means, while the expanded versions are shown on the fifth and seventh lines. (Note that the values shown are not the actual values that would be obtained for this particular network; they are intended merely to illustrate how the process might typically look.)


1
2
3
4
5
6

MEANS
1
2
3
4
5
6
STARTING POSITION

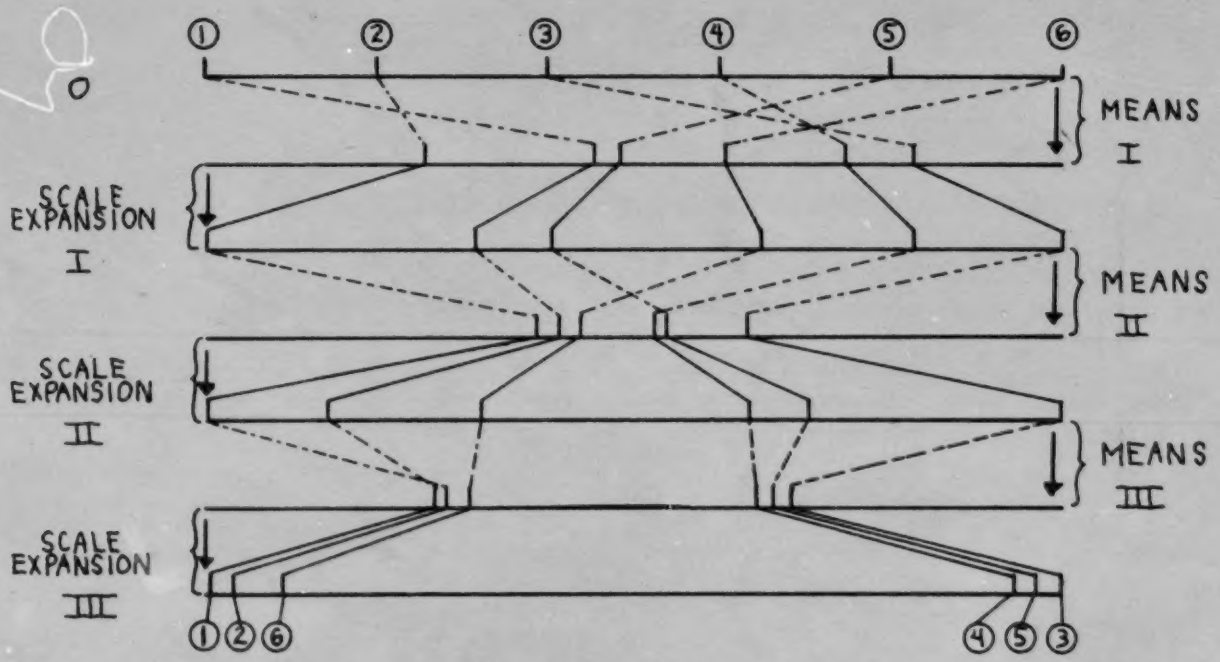

0
SCALE EXPANSION I
SCALE EXPANSION II
SCALE EXPANSION III
MEANS I
MEANS II
MEANS III
1 2 3 4 5 6
1 2 6 4 5 3

FIGURE 4

of the mean. For this reason, the entire process is repeated, using the new locations instead of the original positions used for the first set of calculations. A plot showing how the nodes moved in successive iterations is shown in the bottom half of Figure 4. Between each set of calculations it is necessary to expand the scale of the continuum so that the spread or range which is occupied by the nodes remains N units long. If this is not done, the points will move closer and closer to each other, finally collapsing on a single spot. This is the "scale expansion" referred to in Figure 4.

The formula used for calculating a person's mean is shown here:

$$M' = \frac{\Sigma(wf_i \cdot S_i \cdot M_i)}{\Sigma(wf_i \cdot S_i)} ,$$

where wf_i is the two-step weighting factor described above; S_i is a ratio-level indicator of the strength of the link; and M_i is the old mean of the person to whom the link goes. The summation is done as i goes from 1 to ℓ, where ℓ is the number of links that the individual whose mean we are calculating has.

In the development of this algorithm, different numbers of iterations, different ways of varying relative contributions of wf_i's, S_i's, and M_i's, and different ways of assigning the original M_i's were tried. In general, four to six iterations seemed to be sufficient for any data set that was examined. If nodes are given subject numbers running from 1 to N, where N is the number of nodes, and these subject numbers are used as the first approximation for the M_i's, the process seems to work well for all types of data. In actual tests, when different subject numbers were assigned

to individuals, the solution obtained was identical to the first solution, which indicates that the process is not terribly sensitive to the original positions. Usually, the wf_i's and S_i's are given equal weight, although this has not been tested extensively.

The result of the application of this process is a continuum, N units long, with a scattering of nodes along its length. A sample network, together with the continuum that might result, is shown in Figure 5. This continuum is used as the input to the next stage of the analysis, in which tentative boundaries for groups are drawn.

Drawing the Tentative Boundaries

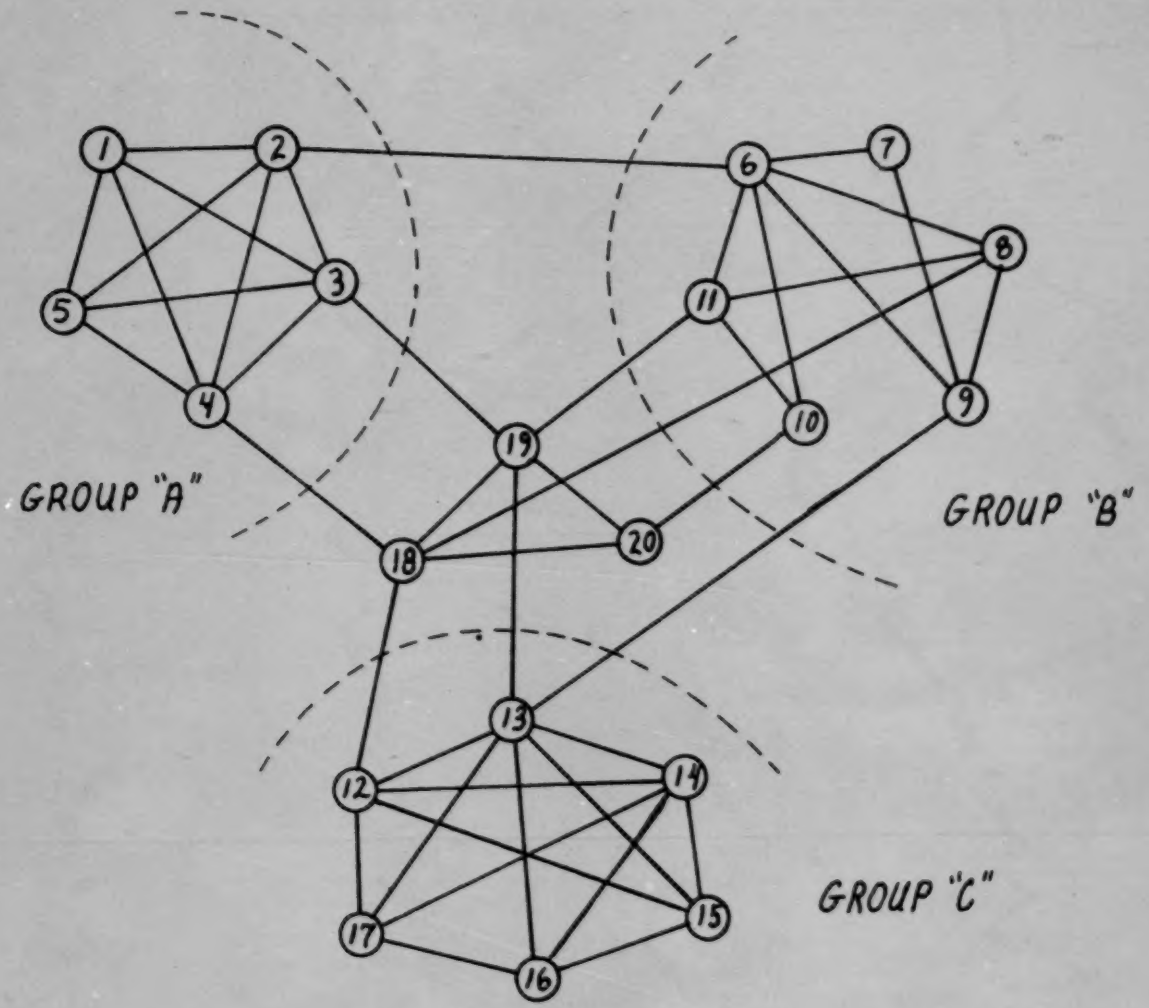
For any human observer, even a casual glance at Figure 5 will be enough to suggest that there are three clusters of nodes. The computer, however, must be told what a cluster looks like, and how to look for one. People probably identify a cluster as an area in which there are a lot of nodes, surrounded by areas in which there are fewer nodes. This is essentially what we have the machine look for.

We will need a plot of the "density" of nodes along the continuum. In order to get such a plot, we construct a "window" and move it along the continuum, counting the number of nodes visible through the window at each point. This is shown at the top of Figure 6. The optimum size of the window, determined by experimentation, appears to be about two units on an N unit line. Windows smaller than this introduce spurious statistical information, while with windows larger than this, group boundaries tend to blur and merge into indistinction. This is shown in Figure 6, where density plots appear for windows of varying widths. The result of moving

FIGURE 5

The top of this figure shows a hypothetical network composed of twenty nodes. Group boundaries are indicated by the dashed lines.

The bottom shows what the final continuum might look like for the network shown in the top. Again, the group boundaries have been indicated by dashed lines.



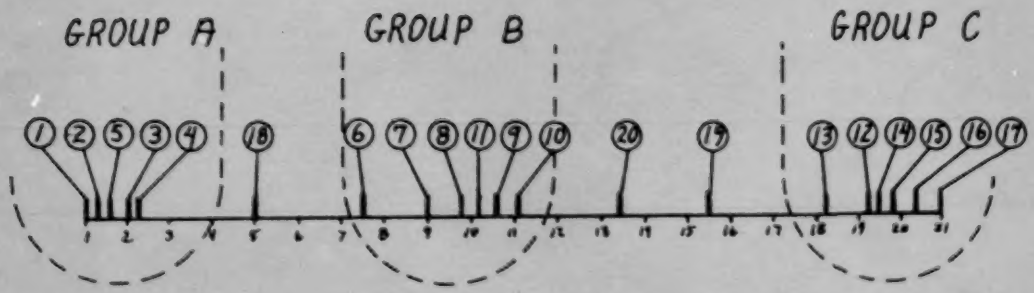


FIGURE 5

FIGURE 6

This figure shows how the density plot is made. The example uses the continuum shown in Figure 5. In the top part, the window is shown, centered successively on the first eight nodes.

The three bar graphs in the middle show the effects of differently sized windows.

On the bottom is shown the refined version of the plot, with numbers of nodes visible to the right of the center of the window plotted above the horizontal and numbers visible on the left of the window plotted below the horizontal.

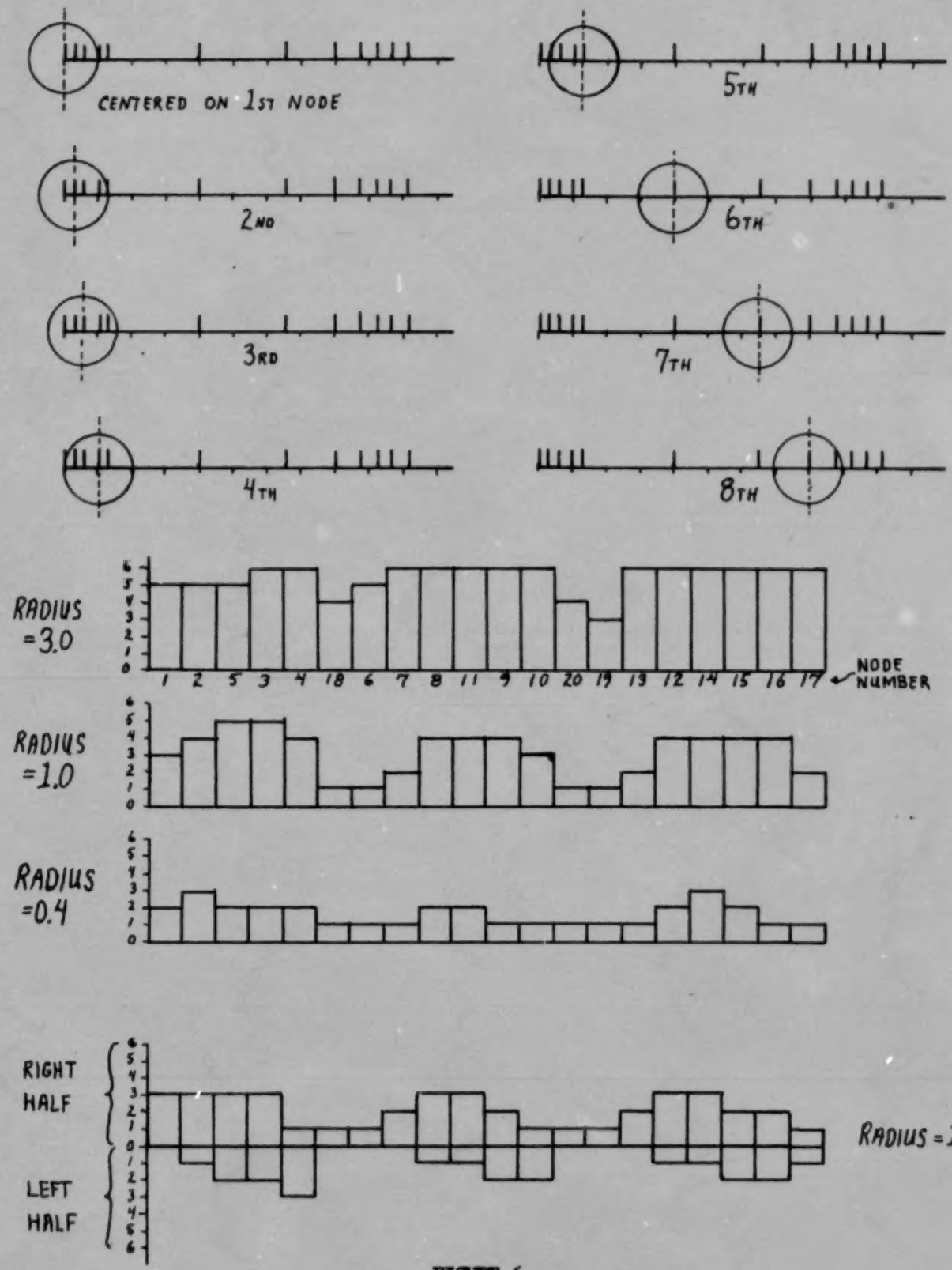

CENTERED ON 1ST NODE
2ND
3RD
4TH
5TH
6TH
7TH
8TH
RADIUS =3.0
1 2 5 3 4 18 6 7 8 11 9 10 20 19 13 12 14 15 16 17
NODE NUMBER
RADIUS =1.0
RADIUS =0.4
RIGHT HALF
LEFT HALF
RADIUS =1

FIGURE 6

the window down the continuum will be a list of densities, with one value for each individual. Such a list could be represented as a bar plot like the one shown in Figure 6.

With this representation, groups will look like mounds, with boundaries between groups being indicated by low points. Although it seems as though this representation would be adequate, there arose problems which lead to an improvement over this simple plot. Although the problems will not be discussed here, the improvement will: instead of just counting the number of nodes visible through the window, two numbers are counted -- the number visible on the right half of the window, and the number visible on the left half. When constructing the bar graph, the number visible on the right half is plotted above the horizontal, while the number visible on the left half is plotted below the horizontal. The result is shown at the bottom of Figure 6.

The final step in this stage is to have the computer draw lines around the groups. This is done by locating spots at which there is a large change as we move from one point on the continuum to the next. If we cound the number of non-overlapping points and divide by the number of overlapping points for each pair of adjacent nodes on the final bar plot, we will have a fairly sensitive indicator of group continuity. This is shown in Figure 7. High values for this ratio will indicate that there is a large change as we move from one node to the next. Low values, on the other hand, will indicate that there is only a small change. If we choose a cutting point and instruct the computer to draw a line whenever the ratio goes above the cutting point, we will have told the computer how to draw the boundaries around groups. If the value of the cutting point is variable,

FIGURE 7

This figure illustrates the boundary-drawing process. The density plot on the bottom of Figure 6 is shown on the top of this figure. The table below the plot shows the number of overlapping points, the number of non-overlapping points, and the ratio of the two numbers for each successive pair of bars on the bar plot.

The ratios are plotted in the graph in the middle of the page. The three dotted lines show the three different cutting points.

Below the ratio plot, the original continuum is shown three times. The first shows the effect of a high cutting point, while the second and third ones show the results for moderate and low values of the cutting point.

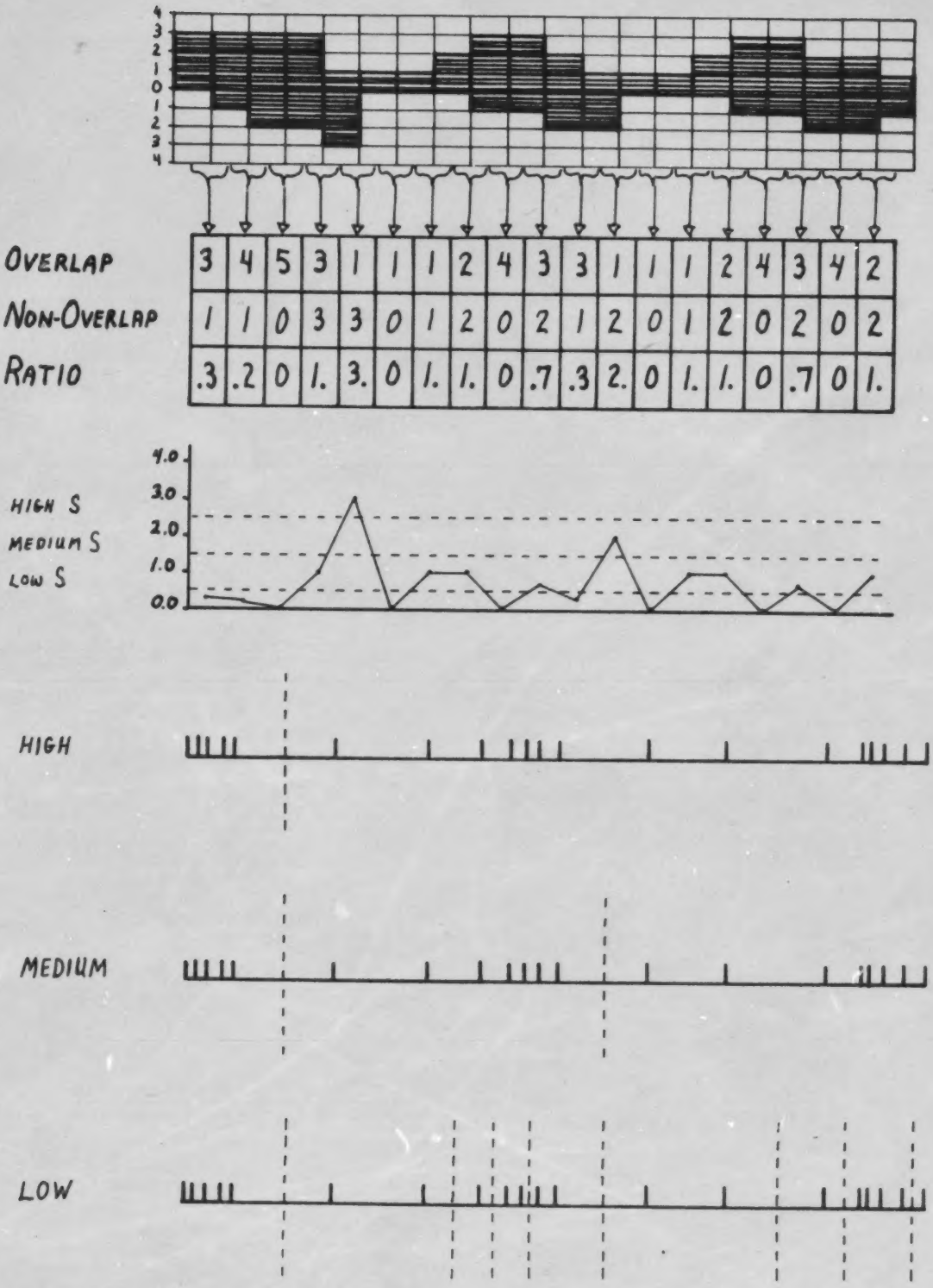

4
3
2
1
0
1
2
3
4
OVERLAP 3 4 5 3 1 1 1 2 4 3 3 1 1 1 2 4 3 4 2
NON-OVERLAP 1 1 0 3 3 0 1 2 0 2 1 2 0 1 2 0 2 0 2
RATIO .3 .2 0 1. 3. 0 1. 1. 0 .7 .3 2. 0 1. 1. 0 .7 0 1.
4.0
3.0
2.0
1.0
0.0
HIGH S
MEDIUM S
LOW S
HIGH
MEDIUM
LOW

FIGURE 7

we can alter the sensitivity of the group spotting routine in either direction. With a window of two units, a cutting point of 1.0 appears to be optimum for most networks. Different values, along with the results, are shown in Figure 7.

This concludes the approximate phase of the analysis. The result of this stage is a list of tentative groups of nodes. The next part of the analysis involves the testing of this tentative solution and any alteration that may have to be done to "clean it up."

Using the Criteria for an Exact Solution

This part of the analysis can be divided into two parts. In the first, individual nodes are tested to see if they meet the relevant criteria for their role in the network. If they do not, the appropriate changes are made. In the second, whole groups are tested for the criteria that are relevant at that level. Again, appropriate changes are made if necessary. We begin with the individual testing, which is very simple.

Individual Testing

First, people not in groups are tested to see if they meet the α-criterion for either liaison or group membership in any group. If any individual does meet the criterion, he or she is reclassified on that basis. If the individual fails both tests, he or she is labelled as "type other."

Second, members of groups are tested to see if they meet the α-criterion for group membership. Again, if the criterion is not met the appropriate changes are made.

Because changes made at any point in time can affect the roles of other people who were tested earlier, the tests are applied twice to make sure that the final classification will be consistent with itself.

Group Testing

In this section, we change our level of analysis to whole groups, rather than separate individuals. The criteria to be tested in this part are the connectiveness and critical link/node criteria. Since the information generated in the testing of the connectiveness criterion is necessary in the testing of the other two, it will be covered first.

The basic device used in the testing of these criteria is the distance matrix, which is constructed for each group. In this n-by-n matrix (n is the number of members in the group), the element in row i, column j gives the number of steps needed to get from individual i to individual j in the group. If there is some finite number in each element of the matrix, the group will be connected. This means that there will be some path from each individual in the group to every other individual in the group. The longest any path could ever be is n-1 steps. A sample network, together with its distance matrix, is shown in Figure 8.

The distance matrix is constructed as follows. A matrix is constructed in which there is a row and a column for each node in the group. All the elements are initialized to zero. Whenever there is a link from node i to node j, a "1" is entered in row i, column j. If the link is reciprocated, a "1" is also entered in row j, column i.

A boolean logic operation which is analogous to raising the matrix to successively higher and higher powers is then performed. Instead of setting the i,j element in the product matrix to the value of the cross

FIGURE 8

At the top of Figure 8 is shown a hypothetical eight-node network. The matrix directly below the network is a binary version of the network. In this matrix, each node has a row and a column. The i,j entry of the matrix is 1 if node i is linked to node j.

The second matrix is the distance matrix for the same network. The entry in the i,j element of the matrix is the number of links in the shortest path from node i to node j.

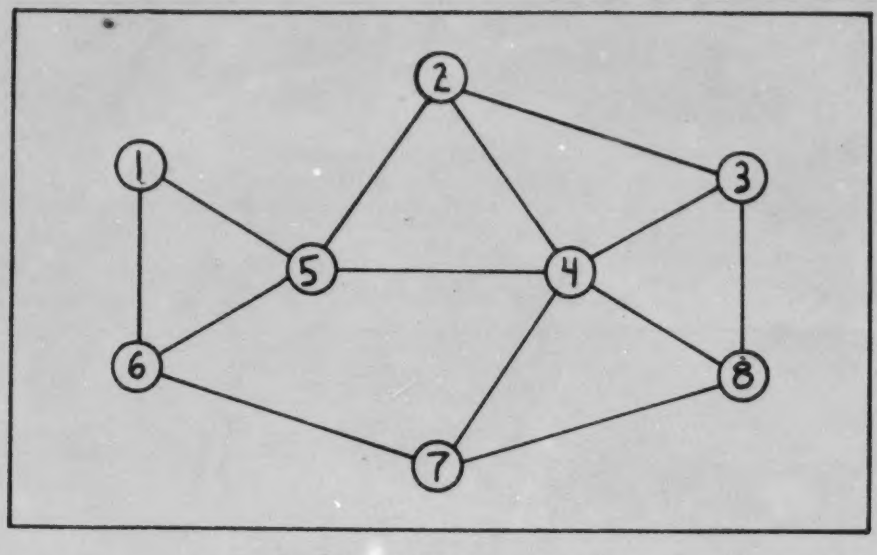


FIGURE 8

BINARY FORM

	1	2	3	4	5	6	7	8
1	0	0	0	0	1	1	0	0
2	0	0	1	1	1	0	0	0
3	0	1	0	1	0	0	0	1
4	0	1	1	0	1	0	1	1
5	1	1	0	1	0	1	0	0
6	1	0	0	0	1	0	1	0
7	0	0	0	1	0	1	0	1
8	0	0	1	1	0	0	1	0

DISTANCE FORM

	1	2	3	4	5	6	7	8
1	0	2	3	2	1	1	2	3
2	2	0	1	1	1	2	2	2
3	3	1	0	1	2	3	2	1
4	2	1	1	0	1	2	1	1
5	1	1	2	1	0	1	2	2
6	1	2	3	2	1	0	1	2
7	2	2	2	1	2	1	0	1
8	3	2	1	1	2	2	1	0

product of the ith row and the jth column, however, the first power on which this value becomes non-zero is used.

The raising of the matrix to higher powers is stopped when one of two conditions obtains either: (a) all off-diagonal elements become non-zero, which implies the group is connected; or (b) when going from any power k to the next power k+1, no entries change value, which implies the group is not connected at level k and will never be connected at any level.

If the group is not connected, it is split into a connected part and all the rest. Each of the two parts is then treated as a separate group, and subjected to all the tests that any group must undergo.

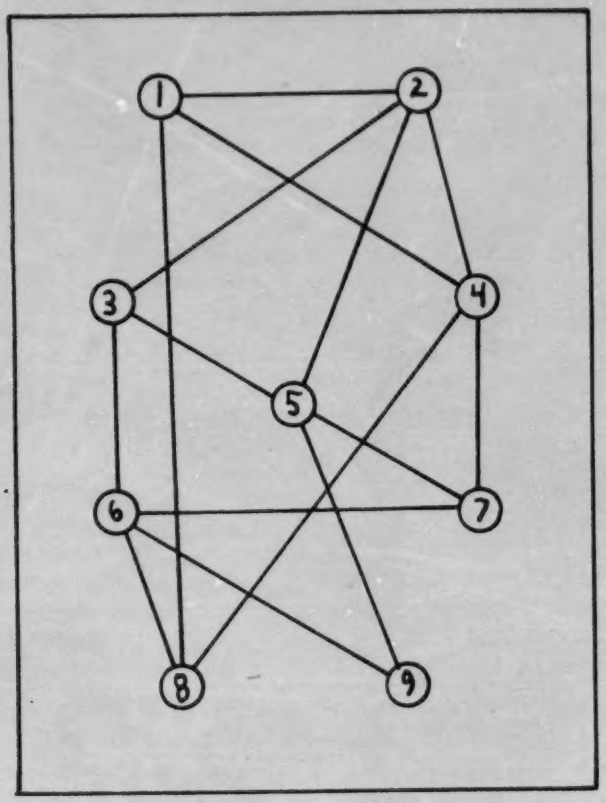
At this point, there are only the critical links/nodes criteria remaining to be tested. These criteria serve as checks against situations like those shown in the bottom half of Figure 9, where two groups have been mistakenly identified as one. This situation is generalized to include situations in which there are any number of multiple groups, connected in some relatively minimal way, which we wish to separate into distinct groups. The occurrence of these confusions is a result of the inelegance of the approximate techniques used in the first half of the analysis. For analytic purposes, it is practical to combine these two criteria into a single rule which says that no subset of some arbitrary size may be removed from a group and cause the group to become disconnected. If there is such a subset, the group will be seen to be "really" two or more groups. As a result of this combination, whenever two groups are joined by a bridge link (a link between members of different groups), one of the nodes of this link will be identified as a liaison. That node will later be tested for the α-criterion of group membership and if it passes, will be returned to the group.

FIGURE 9

On the upper left-hand corner of this figure is shown a hypothetical nine-member network. To the right of this is the distance matrix for that network. The rightmost column of the matrix contains the means of the rows of the matrix. The values in this column are thus the mean number of steps it takes that node to reach all other nodes. The overall mean for the group, together with the standard deviation of the distribution of means, is shown below the matrix.

The network in the bottom left-hand corner is an example of the kind of situation that occurs when two or more groups are identified as a single group. Clearly, Node #5 is a liaison between the two groups. The middle matrix on the right half of the page is the distance matrix for this group. Note the relatively high standard deviation for this group, compared to the one above it.

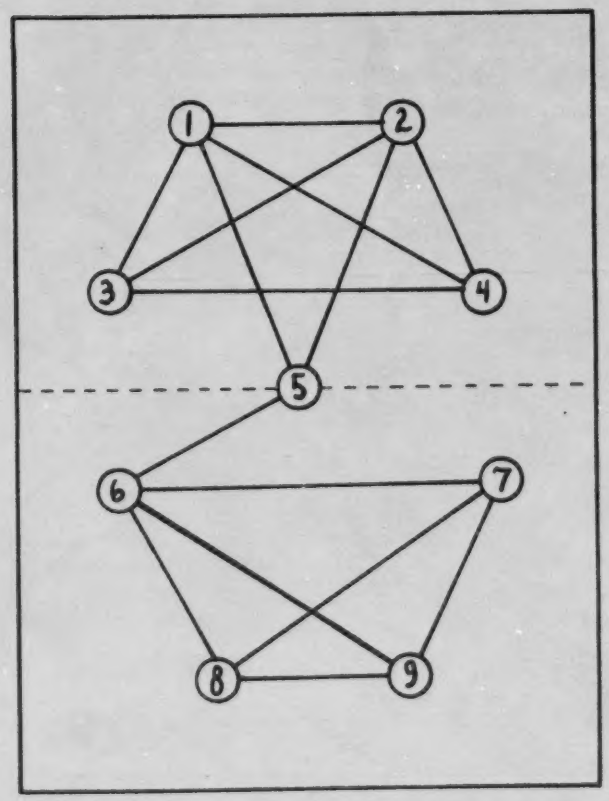
The third matrix was constructed after removing Node #5. Note that there are no values for many of the elements, indicating that the group is no longer connected. The means shown for this bottom matrix are the values that would be obtained if the group were split in two, and the means for each group calculated separately.



	1	2	3	4	5	6	7	8	9	
1	0	1	2	1	2	2	2	1	3	1.75
2	1	0	1	1	1	2	2	3	2	1.62
3	2	1	0	2	1	1	2	2	2	1.62
4	1	1	2	0	2	2	1	1	3	1.62
5	2	1	1	2	0	2	1	3	1	1.62
6	2	2	1	2	2	0	1	1	1	1.5
7	2	2	2	1	1	1	0	2	2	1.62
8	1	3	2	1	3	1	2	0	2	1.87
9	3	2	2	3	1	1	2	2	0	2.0

MEAN=

S.D. =



	1	2	3	4	5	6	7	8	9	
1	0	1	1	1	1	2	3	3	3	1.87
2	1	0	1	1	1	2	3	3	3	1.87
3	1	1	0	1	2	3	4	4	4	2.5
4	1	1	1	0	2	3	4	4	4	2.5
5	1	1	2	2	0	1	2	2	2	1.62 ←
6	2	2	3	3	1	0	1	1	1	1.75
7	3	3	4	4	4	1	0	1	1	2.61
8	3	3	4	4	4	1	1	0	1	2.61
9	3	3	4	4	4	1	1	1	0	2.61

MEAN =

S.D.=

	1	2	3	4	5	6	7	8	9	
1	0	1	1	1	–	–	–	–	–	1.0
2	1	0	1	1	–	–	–	–	–	1.0
3	1	1	0	1	–	–	–	–	–	1.0
4	1	1	1	0	–	–	–	–	–	1.0
5										
6	–	–	–	–	–	0	1	1	1	1.0
7	–	–	–	–	–	1	0	1	1	1.0
8	–	–	–	–	–	1	1	0	1	1.0
9	–	–	–	–	–	1	1	1	0	1.0

"A"

"B"

FIGURE 9

The problem has thus been reduced to one of identifying any critical nodes which may exist in a group. If there is one, it will be the node with the lowest average distance from all other nodes. This is because all paths from nodes in either half of the group to the other half must go through the critical node. The average distance from any node to all the other nodes is given by the average of all the entries in that node's row in the distance matrix. This is illustrated in Figure 9. If there is a set of critical nodes, they will be the nodes with the smallest row means.

The fact that critical nodes have lower row means than the other members suggests that there must be some variation in the row means if there are any critical nodes. We can take advantage of this fact if we only look for critical nodes when there is some variance. It turns out that this leads to a large savings in terms of computation time. This is because of the way we test for critical nodes.

To check a node to see if it is critical, we remove it from the group and re-calculate the distance matrix. If, as a result of the removal, the group becomes disconnected, we have found a critical node. If the group is still connected, we try the next candidate -- the node who, of all the remaining nodes, has the smallest row mean. We will stop this process after taking out some percentage of the original group members (usually 10% is enough to "catch" all the critical nodes) if the group continues to remain connected. If this happens, we put all the removed nodes back into the group.

It is easy to see that there is a lot of work involved in the searching for critical nodes. This is why the heuristic device of checking the

variance of the row means is so important. In every network that has been examined so far, this heuristic has worked correctly. That is, it did not prevent any critical nodes from being found. Similarly, the approach of looking at nodes with the lowest row means always finds the critical nodes. The optimum value to use as a cutting point for the variance test seems to be about 0.3. Whenever the standard deviation of the row means exceeds this value, there is likely to be a critical node. Whenever the standard deviation is less than this value, there is not.

After all groups have passed these tests, the obtained classification of nodes to groups and other roles will be exact. At this point, various indices may be calculated and the results tabled in any convenient manner. A flow chart of the algorithm is shown in Figure 10.

II. NEGOPY: THE NETWORK ANALYSIS PROGRAM

In this section, Negopy, the Network Analysis Program, is discussed. Since there are five parts in the analysis, the discussion is divided into fiv e parts: describing the data, preparing the data for group detection, initial group detection, applying the formal criteria, and printing the results. In each part we will briefly review the relevant parts of the algorithm, discuss the parameters by which the user may control the computer (there are 45 parameters which control the operation of the network program, much as the knobs on a radio control the way it works), and describe the output produced by this part of the analysis.

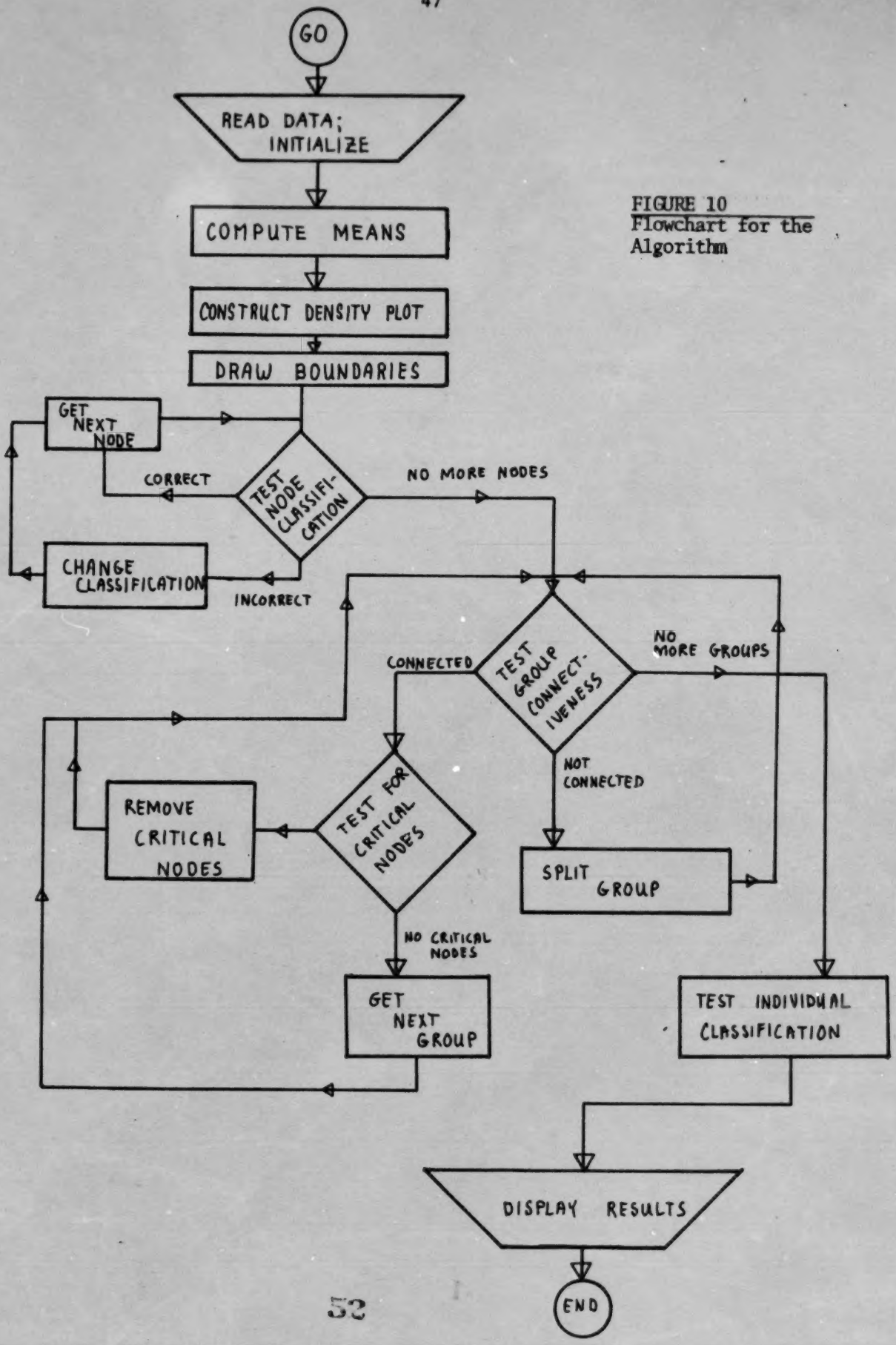


FIGURE 10
Flowchart for the Algorithm

1. Data Description

A. Review

In this stage, the computer will read in the data according to format statements provided by the user. Any transformations needed to approximate ratio scaling of link strength is done here.

B. Parameters

P01 -- # OF NODES DEFAULT = 0
MAX = 4095

The first parameter that is specified by the user is P01. It tells the computer how many nodes there are in the network. It also tells the computer what the highest legal subject number is. In other words, the computer expects that nodes are numbered from 1 to N, where N is the number of nodes. If this is the case, N will be the highest legal subject number. If this is not the case, if subjects are not numbered 1 to N, the value for this parameter must be set to the largest legal subject number. For example, say there are 100 nodes, numbered from 101 to 200. Even though there are only 100 nodes, P01 must be set to 200. In this case, the computer will assume that there are 200 nodes. Naturally, nodes 1 to 100 will have no links. They will be identified as isolates type one. If P01 is set to 100, all links to or from nodes having numbers greater than 100 will be rejected.

This is important: all links to or from nodes having numbers greater than the value of P01 will be rejected by the computer.

P02 -- # OF LINKS DEFAULT = 0
MAX = 32,767

This is an estimate of the number of links. It is used by the computer to allocate memory space. This parameter should always be set to

a value about 200 higher than the number of expected links. If the actual number of links exceeds this number, only the first P02 will be read. The excess links will be ignored.

P03 -- RECIPROCAT DEFAULT = 1

If this parameter is set to zero, the computer will drop all unreciprocated links. If it is set to one, the computer will force reciprocation by adding the "missing halves" of all unreciprocated links. In this case, the computer will carry information which tells if the link is reciprocated, unreciprocated, or "added." This information is printed at the appropriate times and will be discussed in the sections on how to interpret the output.

P04 -- DIRECTION DEFAULT = 0

If this parameter is left at the default value of zero, the computer assumes links are non-directed. That is, if there is a link from A to B, there is also one from B to A. (Links are always "from" the person reporting the link.) If the parameter is set to one, the computer assumes links are directed: a link from A to B does not imply a link from B to A.

P04 is related to P03. If P03 is set to 0 (dropping non-reciprocated links), P04 should be left at the default value of 0, since all links are reciprocated and, thus, bi-directional. If P03 is set to one, however, P04 can be set to either zero or one. If P04 is set to the default value of 0, the computer will assume that unreciprocated links are evidence of measurement error and will add the missing halves. If P04 is set to one,

however, the computer will not make this assumption. Although it will still add links to force reciprocation, it will not use the "added" links in calculations where this would violate the assumption of directionality. In this case, the computer will either print only a single value appropriate to the directed or else print two values -- the first appropriate to non-directed links and the second to directed links. The computer will always print information regarding this decision.

P07 -- DATA UNIT DEFAULT = 40

If this parameter is left at the default value of 40, the computer will attempt to read the data from a file which has the local name "data." If data are on a permanent file, the file must be attached with the local name "data" before executing the program. If the parameter is set to 60, the computer will attempt to read the data from the input stream, which will be cards if the job is submitted through batch. If the job is submitted from a remote terminal, the data can be either submitted as a file, with P07 = 40, or else they can be appended to the input stream as card images, with P07 = 60.

If P07 is set to the incorrect value, the program will terminate with a message stating that no links have been read.

P31 -- # WEIGHTS DEFAULT = 1
MAX = 2

If only a single indicator of link strength is used, P31 should be left at the default value of one. If two indicators are used, P31 must be set to two. For information on combining indicators, see the section on the link-weighting formula, page 12-15.

P08 -- # OBSV/CARD DEFAULT = 1
MAX = 10

This parameter indicates the maximum number of observations (links) that may appear on a card or card image. Links are formatted as shown below:

R#	L#1	L#2	L#3	...	L#n

The first value must be the respondent number -- the subject number of the person reporting the link. This value is followed by the fields describing the links. These fields must contain either two or three numbers. They will either look liks this: | C# | X | or this: | C# | X | Y | . Here, the "C#" refers to "contactee number" -- the subject number of the person to whom the link goes. The "X" or "X" and "Y" refer to indicators of the strength of the link.

There may be up to ten links per card. If a person has more than P08 links, the first P08 may be put on one card and the others on other card(s). If more than one card is needed, the second (and subsequent) cards must be formatted the same as the first. That is, the respondent number appears in the same columns on later cards as it did on the first one. There is no limit to the number of links a node may have. If a node has less than P08 links, the rest of the card is left blank.

P08 should be set to the maximum number of links that appear on any card (i.e., if there are never more than 9 links on a card, it should be set to 9, etc.).

P09 -- NAME-WIDTH	DEFAULT = 0 MAX = 2

The program allows up to twenty columns of information to be read in for each node. Since this is most often used to read in a list of names of nodes, the list is called a "namelist." The information associated with each node is printed whenever that node is referred to. If there is no namelist, P09 should be set to zero. If ten columns or less are used for each name, P09 is set to one. If up to twenty columns are used, P09 is set to two. The format of the namelist is discussed in the section on running the program.

P10 -- LOW WEIGHT	DEFAULT = 1
P11 -- HI WEIGHT	DEFAULT = 1 MAX = 255

P10 specifies the lowest legal strength a link may have. Links with strength less than P10 are dropped. P11 specifies the highest legal strength a link may have. Links with strength higher than P11 are dropped. These limits refer to strengths after calculations by the link weighting formula which appears below.

P15 -- EXPONENT	DEFAULT = 1 MAX = 4
P37 -- CONS	DEFAULT = 0
P38 -- MX	DEFAULT = 1
P39 -- MY	DEFAULT = 0
P40 -- CCX	DEFAULT = 0
P41 -- CCY	DEFAULT = 0
P42 -- MCPK	DEFAULT = 0

These parameters are all used in the link weighting formula shown on the next page.

These parameters govern the link weighting formula, which is shown below.

$$STRENGTH=[(CONS+MX\cdot Xweight+MY\cdot Yweight)+((CCX+Xweight)\cdot(CCY+Yweight)\cdot MCPK)]^{exp}$$

This formula provides a way of performing a variety of transformations on the strength indicators. The "Xweight" refers to the first indicator of the strength of the link. The "Yweight" refers to the second, if there is a second one (i.e., if P31 = 2). If P31 = 1, the value of Yweight will always be zero.

The link weighting formula can be broken down into two parts. In the first, or linear, part, the Xweight is multiplied by MX, or P38. The Yweight is multiplied by MY, or P39. These two products are then added to CONS, P37, to form a single value. This part is used either to form a simple linear combination of two weights or to reverse the scale on a weight.

> An example of linear combination. Say we have two indicators--time spent in face-to-face conversation and time spent on the telephone. If we decide that face-to-face counts twice as much as telephone, we would set MX to 2.0, MY to 1.0, and CONS to 0.0. We would also set CCX, CCY, and MCPK to 0.0 and EXP to 1.0.

> An example of scale reversal. Say we have frequency of interaction as our only weight. It has been coded as 1 = several times a day, 2 = several times a week, 3 = several times a month. We wish to reverse the scale so that 1 = several times a month and 3 = several times a day. To do this we would set CONS to 4.0, MX to -1.0, and MY to 0.0. We would also set CCX, CCY, and MCPK to 0.0 and EXP to 1.0.

The second part of the link weighting formula is the cross product part. It allows a product to be formed between the Xweight and the Yweight.

An example of cross products. Say we have two indicators of strength: frequency (X-weights) and importance (Y-weights). They have been coded as shown below and we wish to combine them as shown so that:

FREQUENCY	IMPORTANCE
1 = once/month	1 = cricial
2 = once/week	2 = highly
3 = once/day	3 = moderately
4 = more than once/day	4 = slightly

FREQUENCY	IMPORTANCE 4	3	2	1
4	8	16	24	32
3	6	12	18	24
2	4	8	12	32
1	2	4	6	8

An interaction that is of crucial importance and that happens several times a day is weighted as 32, and one that is only slightly important and that happens only once a month is weighted as 2.

First, we have to reverse the importance scale. We do this by setting CCY to -5, which gives us a new importance range of -4 to -1. Then we set CCX to 0 because we want to keep the frequency coding intact. Finally, we set MCPK to -2. This (a) doubles the values we get for the products, and (b) reverses the sign caused by the reversal of the importance scale. We would set CONS, MX, and MY to 0.0 and EXP to 1. Thus,

$$\text{STRENGTH} = [((0)+\text{X-weight})*((-5)+\text{Y-weight})*(-2)]^{1}$$

The entire quantity in the linear part is added to the entire quantity in the cross-product part. At this time, P15, EXP, may be used to raise the sum to the second, third, or fourth power. If this is not needed, P15 is left at the default value of 1. P15 may take no values other than 1, 2, 3, or 4.

Because the maximum value of P11 is 255, the upper limit to the final strength is also 255. That is, the strength of a link may never exceed 255. If a scale for duration is used, coded in minutes, and the range of values is 1 to 1000, the range could be reduced by se-ting MK to 0.25. This would give a range of 0 to 250, which is acceptable.

The final value for strength is expressed as an integer. Thus, values are truncated (rather than rounded) to the next lowest integer. A fractional value less than one -- say 0.75 or 0.99 -- would be truncated to zero.

P06--# RAW PRINT DEFAULT = 10

The computer prints both the raw data and the final strength values for links until it has read P06 acceptable links. This means that if only one out of every four links is acceptable after transforming according to the link weighting formula, the computer will print the first forty raw links as well as the first ten good final links. This is useful in checking the link weighting formula for correctness.

P34 -- MEAN SIRST DEFAULT = 0

If this parameter is set to one, the mean value for both halves of each reciprocated link will be calculated, printed out, and used to replace the original values. If it is left at the default value of zero, the mean will be calculated and printed, but the original values will be retained.

C. Output

The program begins its printout with a list of the parameters and their settings. A portion of this list is shown on the top of the next page. If a parameter is left at a default value, the word "DEFAULT" will appear by that parameter. If the user supplies a value, the word "USER" will appear instead. This list is followed by the namelist, if there is one. In this namelist, the "names" will appear in numerical order, by subject number. Following the namelist will be the first P06 good links.

After all the links have been read into the computer, links will either be dropped or added, depending on the value of P03. The computer indicates how many links were dropped or added in this process.

```
----NODE NO     2       ID=     HARRY
  IS CONNECTED TO
       NODE NO.      3    R  STRENGTH= 5  STRENGTH IN= 7  MEAN STR.= 6  DISCRP= -2    ID=     MARY
       NODE NO.      4    R  STRENGTH= 8  STRENGTH IN= 5  MEAN STR.= 6  DISCRP=  3    ID=     VIKI
       NODE NO.      8    U  STRENGTH= 8                                               ID=     ADAM
       NODE NO.     10    R  STRENGTH= 7  STRENGTH IN= 5  MEAN STR.= 6  DISCRP= -1    ID=     EDWIN
       NODE NO.     12    A  STRENGTH= 6                                               ID=     SAM

                DISCREPANCY TABLE

                    SUM        MEAN
          ACTUAL      3.        1.00
          ABSOLUTE    7.        2.33

                THERE ARE    3. RECIPROCATED LINKS

                                STRENGTH TABLE

                                OUT                                IN                                TOTAL
RECIPRCTD     S=  20.  N= 3.  M=  6.67     S=  17.  N= 3.  M=  5.67     S=  37.  N= 6.  M=  6.17
UNRECPRCTD    S=   8.  N= 1.  M=  8.00     S=   6.  N= 1.  M=  6.00     S=  14.  N= 2.  M=  7.00
  TOTAL       S=  28.  N= 4.  M=  7.00     S=  23.  N= 4.  M=  5.75     S=  51.  N= 8.  M=  6.38
```

FIGURE 11 -- LINK LIST FOR NODE # 2

At the top is a list of all links involving node #2, the node used for this example. The "R", "U", or "A" indicates whether the link was reciprocated (two-way), unreciprocated (outgoing), or added (incoming). The strength of the link as reported by the respondent follows, along with the strength as reported by the other person involved, and the difference between the two strengths. (Discrepancy information only appears for reciprocated links.) The discrepancy table has rows for actual values of discrepancies and for absolute values (ignoring the sign). Finally, the strength tables break down links into outgoing and incoming, as well as reciprocated and unreciprocated.

```
PARAMETER NO. 1   (N OF NODES)  VALUE =   50  **USER**
PARAMETER NO. 2   (N OF LINKS)  VALUE =  400  **USER**
PARAMETER NO. 3   (RECIPROCAT)  VALUE =    1  DEFAULT
PARAMETER NO. 4   (DIRECTION )  VALUE =    0  DEFAULT
PARAMETER NO. 5   (N OF ITERS)  VALUE =    4  DEFAULT
PARAMETER NO. 6   (N RAW PRNT)  VALUE =   10  DEFAULT
PARAMETER NO. 7   (DATA UNIT )  VALUE =   40  DEFAULT
PARAMETER NO. 8   (NOBSV/CARD)  VALUE =    8  **USER**
PARAMETER NO. 9   (NAME-WIDTH)  VALUE =    1  **USER**
PARAMETER NO. 10  (LOW WEIGHT)  VALUE =    1  **USER**
PARAMETER NO. 11  (HI WEIGHT )  VALUE =   10  **USER**
PARAMETER NO. 12  (INOPERATIV)  VALUE =    0  DEFAULT
PARAMETER NO. 13  (DNSTY HIST)  VALUE =    1  DEFAULT
PARAMETER NO. 14  (SCAN RDIUS)  VALUE =  200  DEFAULT
      .               .             .          .
      .               .             .          .
      .               .             .          .
```

A portion of the parameter list

Next comes the first major part of the output: the link list. In this list all the links are displayed, beginning with the links for Node #1, then Node #2, and so on. A part of this list is shown in Figure 11. If a link is reciprocated, there will be an "R" after the node number to whom the link goes. Similarly, there will be a "U" for an unreciprocated link and an "A" for added links. In the example shown in Figure 11, Node No. 2 listed Node No. 8, but Node No. 8 did not list Node No. 2. On the other hand, Node No. 12 listed Node No. 2, who did not list Node No. 12.

To the right of the reciprocation indicator is the word "STRENGTH=" followed by the strength of the link as reported by the respondent. If the link was reciprocated, the words "STRENGTH IN=" will appear, followed by the strength as reported by the other person. In the example, Node No. 2 reported a link to Node No. 3 with a strength of 5. Node No. 3, however, reported a link to Node No. 2 with a strength of 7. The number after the words "MEAN STR. =" is the mean of incoming and outgoing links. The number after "DISCR=" is the discrepancy between incoming strength and outgoing strength (calculated as outgoing minus incoming). Finally, "ID="

is followed by the "name" of the person to whom the link goes.

After all the links for the node are listed, there appear two tables. The first is a "DISCREPANCY TABLE," where the differences between outgoing and incoming strengths are analyzed. Obviously, this can only be done for reciprocated links. There are two rows in this table. The first refers to actual values, while the second refers to absolute values (ignoring the signs). There are also two columns. In the "SUM" column the discrepancies have been added together into a total. In the "MEAN" column the sum has been divided by the number of reciprocated links, which appears to the right of the table.

If the values in the "ACTUAL" row are small, this could be due to one of two situations. Either all the discrepancies are small, or else the discrepancies are large, but the positive ones are balanced out by the negative ones. If the values in the "ACTUAL" row are large, this means the node consistently differed in estimating the strengths of links from those it was linked to, and that the difference was usually in the same direction. In other words, such a node could be said to be an overestimator if its "ACTUAL" values were large and positive, or an underestimator if its values were large and negative. Thus, the "ACTUAL" values are useful to see if there is any systematic bias in the direction of the discrepancy.

The values in the "ABSOLUTE" row, in contrast, reflect only the magnitudes of the discrepancies. Nodes with low values here reported strengths that were very close to the strengths reported by the nodes

they were linked to. Nodes with high values, on the other hand, were simply not in agreement with the nodes they were linked to.

Following the discrepancy table is the "STRENGTH TABLE." The numbers under "OUT" refer to strengths as reported by the node whose number appears at the top of the link list (Node No. 2 in the example). The numbers under "IN" refer to strengths as reported by the nodes linked to the respondent. The numbers under "TOTAL" refer to all the strengths combined. The first row refers to reciprocated links only, the second to unreciprocated links only, and the third to all links combined. There are thus nine cells in this table.

In each cell are: the sum of the strengths for the appropriate links (after "S="); the number of links (after "N"); and the mean strength (after "M="). Whenever there is a blank entry, with simply a ".", there were no links for that cell and the value is zero.

With this table we can make statements like the following: "On the whole, for Node 2, outgoing links (i.e., links reported by Node 2) were stronger than incoming links. However, the difference was greater for unreciprocated links, Δ = 8.0-6.0 = 2.0, than for reciprocated links, Δ = 7.0-5.75 = 1.25." or "Reciprocated links with Node 2 were stronger than unreciprocated links."

There is a discrepancy table for each node that has some reciprocated links. When there are no reciprocated links, the discrepancy table and the top row of the strength table do not appear. In addition, if the node has no links the computer prints "HAS NO LINKS" after the node number, and skips to the next node.

After all the links for all the nodes have been listed, there appears a small table labelled "RECIPROCATION ANALYSIS." Here the entire set of links are broken down into types, according to whether they were reciprocated (two-way), unreciprocated (outgoing), or added (incoming).

Finally, there is a section headed "STRENGTH DISTRIBUTION ANALYSIS" where the distribution of strengths is analyzed. After a self-explanatory description of the range of strengths appears a histogram where each legal strength value has a row and the length of the bar in that row is proportional to the number of links with that strength. If the longest row has more than 100 links, the lengths of all rows will be divided by 10. If the longest row would still be over 100 X's, the longest row has less than 100 X's. The exact scale factor is indicated at the top of the table. An example of this histogram is shown in Figure 12.

2. Preparation for Group Detection

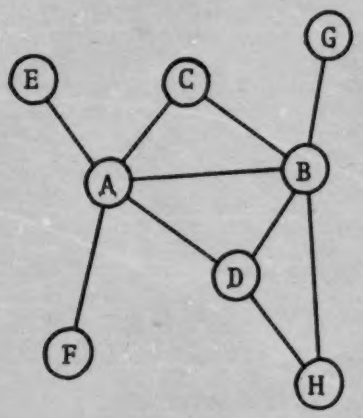
A. Review

The computer begins this segment by identifying all the non-participants. It separates them from the participants since they are not used until the final stage of analysis. Following this, the computer treats each link as a vector going from the respondent to the contactee. Links are weighted according to their strength and the within-group weighting factor. The mean vector is calculated for each node and used to relocate the node for the next round of means. The whole process is repeated a number of times.

B. Parameters

P44 -- 2-STEP WT DEFAULT = 1.00

This parameter is used to adjust the contribution of the within-group weighting factor. This factor, used as an indicator of the probability that the link is a within-group link, is calculated as the number of two-step links connecting the nodes plus one. In the example to the left, the weighting factor for the A-B link would be three, because of the indirect links connecting A and B through C and D.



The formula used to calculate the new mean for node j is:

$$MEAN'j = \frac{S_i M_i (wf_i \cdot P44)}{S_i (wf_i \cdot P44)}$$

(Discussed also on page 29)

By adjusting P44, the influence of the within-group weighting factor can be varied. P44 is usually set to 1.0 Lower values are not recommended; the effect of higher values has not been tested.

P05 -- N OF ITERS DEFAULT = 4

This parameter specifies the number of iterations to perform -- the number of times means are to be calculated. Four is usually sufficient. For datasets with very large numbers of links, it may be necessary to use six or seven. The best number is determined by experimentation.

C. Output

Output for this segment begins with a list of non-participants as the computer finds them. Isolates will be located first, followed by tree nodes. After all non-participants are located, the computer removes links to non-participants from the link lists of participants. This is done to simplify the rest of the analysis. There is no more output from this segment unless there are not enough participants to continue. If this is the case, the program will terminate here with a message.

3. Initial Group Detection

A. Review

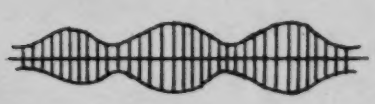
The result of the preceding segment is a continuum with nodes scattered along its length. In this segment, the computer examines this continuum by moving a "window" down the continuum, counting the number of nodes appearing in both halves of the window. These numbers are displayed in a density histogram, and analyzed to locate boundaries of groups. A group boundary is drawn whenever the transition from one location of the window to the next causes a shift in density values that is larger than the sensitivity parameter.

B. Parameters

P14 -- SCAN RADIUS DEFAULT = 200

This parameter governs the radius of the window that is used to calculate densities along the continuum. The value of P14 is actually one hundred times the width of the window, where the continuum is N units long, where N is the number of nodes in the network. A value of 200 has been found to be optimal for most networks. If the density histogram

appears to be "blurry" with gradual, smooth transitions from group to group (i.e., as in "A" below), the radius should be decreased to a smaller



A. A blurry histogram

B. A sharp histogram

value. If the value is made too small, however, the histogram will be very thin and spotty, as shown in "C" below. In this case there will probably be no groups identified.

C. A thin spotty histogram

P13 -- DNSTY HIST DEFAULT = 1

If P13 is set to zero, no density histogram will be printed. It is recommended that this not be done, since the histogram provides valuable information in a relatively compact form.

P23 -- GRP-SNSVTY DEFAULT = 100

This parameter is discussed on page 35. Whenever the computer detects a group boundary, it draws a line across the histogram. If it is felt that the computer is not splitting the continuum into enough groups, the sensitivity can be raised to a higher value. Similarly, if the computer

is making too many splits, cutting groups apart, the value can be lowered. The value should not be changed by more than twenty units at a time, as it has drastic effects on the pattern-recognition routines that look for groups.

C. Output

The output from this section consists entirely of the density histogram that has been discussed above. A segment of a density histogram is shown below. The numbers along the left side are node numbers. The

```
---------------------------------------------------------------------
  2      1                          .XXXXX
  6      1                         X.XXXX
  4      1                        XX.XXX
  8      1                       XXX.XX
 12      1                      XXXX.X
 10      1                     XXXXX.
---------------------------------------------------------------------
 14      1                          .XXXXXX
  1      1                         X.XXXXX
 13      1                        XX.XXXX
  3      1                       XXX.XXX
  9      1                      XXXX.XX
 11      1                     XXXXX.X
  7      1                    XXXXXX.X
---------------------------------------------------------------------
  5      0                         X.
 21      0                           .XX
 26      0                         X.X
---------------------------------------------------------------------
 16      1                        XX.XXXXXX
 19      1                         X.XXXXX
 17      1                        XX.XXXX
 42      1                       XXX.XXXX
 27      1                      XXXX.XXX
 45      1                     XXXXX.XX
 46      1                    XXXXXX.X
 23      1                     XXXX.X
---------------------------------------------------------------------
 24      0                         X.
  .
  .
  .
```

numbers immediately to the right of the node numbers (the zeros and ones) refer to the "scanning mode" of the computer at that point on the continuum (which has been turned vertically so the top is the left end and the bottom is the right end). When the computer is working on a group, it is in group mode and there will be a "1" to the right of the subject number. When the computer is between groups, there will be a "0" instead. The horizontal lines are boundaries of groups. The numbers of "X's" in a row equals the number of nodes visible through the window when it is centered over the node whose number appears on the left end of the row.

At the bottom of the histogram is the phrase "NGRP FROM GROUP IS," followed by a number. This is the number of groups found by the computer in its analysis of the density histogram. If there are no groups, the computer will stop here with a message.

4. Application of Formal Criteria to the Tentative Solution

A. Review

In this segment the computer applies the formal criteria to the tentative solution provided by the third segment. First, individual nodes are tested to see if their role classification is consistent with the role definitions. This is done by computing the appropriate proportions of linkages with group members and comparing the results to the criterion levels as specified.

If there are enough type "others" at this point, the computer attempts to construct additional groups from these nodes.

The computer then proceeds to test each group to see if it meets the criteria for groups. Most of this work is done with the aid of the

"distance matrix" which is discussed on pages 39-42. If a group is not connected, it is split apart into a connected part and all the rest. If the variance of row means in the distance matrix is high enough, the computer tries to split apart the group by removing critical nodes. If the computer succeeds, it makes two groups out of the remaining members. It then applies all the relevant tests to those new groups. If the computer does not succeed at splitting apart the group, it returns all the nodes it took out in the process of trying to split the group.

Finally, after all groups have been tested, the computer applies the appropriate criteria to the nodes which remain outside groups.

B. Parameters

P36 -- PERW DEFAULT = 50.01

This is the α-percentage used in all the criteria. It is discussed on page 5 . The default value of 50.01% is as low as possible for unambiguous classifications of nodes. Because higher values have not been tested extensively, it is not possible to say how they affect the operation of the program.

P22 -- MIN SPLIT	DEFAULT = 12 MIN = 5
P24 -- SPLIT DEV	DEFAULT = 30 MIN = 5

The computer will attempt to split any group that has at least P22 members by removing critical links, if the standard deviation of the row means in the distance matrix is greater than the value of P24 divided by 100. That is, if both P22 and P24 are left at their default values, the computer will attempt to split any group having 12 or more members whenever the standard deviation of that group row means exceeds 0.30.

P43 -- DROP-SPLIT DEFAULT = 0.10

This parameter specifies how many nodes may be removed in attempting to split a group. If P43 is set to a number greater than 1.0, the number is the largest number of nodes that will be removed. If P43 is set to a fraction between 0.0 and 1.0, up to this proportion of the group will be removed. Thus, a value of four means that up to four nodes will be removed. A value of 0.1 means that up to 10% will be removed.

C. Output

The output of this segment begins with a brief reporting of the results of the tests of individuals. When a node is reassigned to a different role, the new role is indicated. After this has been done, the word "GRZAP" appears, followed by a list of the tentative groups (they have not yet passed the group criteria testing).

NOTE: The groups at this point are numbered from 1 to N, where N is the number of groups. This numbering may change as the groups are tested. In fact, after testing there may be "empty groups" -- numbers for which there are no groups. For example, there may be groups numbered 1, 2, 4, 6, and 9. There are no groups numbered 3, 5, 7, or 8. There are only five valid groups. (This is inconvenient, I know. I will fix it someday, maybe.)

The computer now begins to apply the criteria to the groups. The first step is to construct a distance matrix. Because this is done in a routine called "CONNECT," the phrase "NOW ENTERING CONNECT FOR GROUP X" appears at the top of the page. Immediately under this is a line that tells whether directed or non-directed links are assumed. (Non-directed links implies a symmetrical distance matrix.)

The computer constructs a binary matrix, where the i,j entry is "1" if there is a link from node i to node j. All other entries are zero.

The number of one-step links is printed out. The computer then raises the matrix to higher and higher boolean powers, where the result after each power is to put the number of the power in the entries corresponding to pairs of nodes that can reach each other in a number of steps equal to the number of the power. Thus, on the second power, a "2" is placed in the i,j entry of the matrix if node i can reach node j in two steps (i.e. with one intermediate node). At each power the number of additional connections that are made is printed.

This process stops when either (a) all pairs of nodes are connected; or (b) no new connections are made when going from one power to the next, which implies that no new connections would ever be made by going to higher powers. The computer prints our the highest power used, together with the result (i.e. "CONNECTED AT LEVEL 2" or "CONNECTING HAS STOPPED AT LEVEL 4".).

The computer then prints out the distance matrix. A matrix assuming non-directed links is always constructed, regardless of the value of P04. This matrix is used for all testing. However, when P04 is set to 1 (directed), an additional matrix is constructed with directional links. This matrix may be asymmetrical. When this matrix appears, it is identified as having used directed links.

For large groups, the entire matrix might not fit on a single page. If this happens, the matrix is printed in strips. To reconstruct the matrix, simply put the strips together.

After the matrix is printed there is a section headed by the phrase "ANALYSIS OF DISTANCE MATRIX". Here there is a table with six columns. The first column is the node number. Column Two has the sum of all entries

in that node's column in the matrix. Column Three has the mean. The mean is calculated by dividing the sum by N-1, where N is the number of nodes in the group. The "-1" is because the elements on the main diagonal are not counted. Rows Four and Five have row totals and means. Column Six only appears if namelists are used, and it has the node's "name."

Below this table is the group column average -- the average of all the numbers in Column Three -- and the standard deviation of the column averages for the members in the group. It is this last number that is used to decide whether or not to try to split the group.

Should the computer decide to try to split a group, the whole process is recorded for later use. When a group is split, there will be at least two matrices printed -- one before splitting and one after splitting. The second one should be used, as the first one is now obsolete.

After the last distance matrix and analysis table, there is a list of nodes not in groups, together with their final classifications (this list does not include non-participants).

5. Final Results and Control of Output

A. Review

In this section, we review the parameters that control the output, along with the tables that summarize the results of the analysis.

B. Parameters

P17 -- FILE OUTPUT DEFAULT = 0

If P17 is set to one, a file called 'PUNCH" will be generated after execution. This file contains all the list structures which represent the network and may be used as input to subsequent analysis programs. For more information about this file, write the author.

P18 -- PRINTO SUP DEFAULT = 0

If this parameter is set to one, the list of links at the beginning of the program will be suppressed.

P19 -- GRID SUP DEFAULT = 1

If this parameter is set to zero, the group structure will be printed out at several intermediate stages in the analysis. This information is useful for tracing the groups as they go through the various tests.

P20 -- GROUP SUP DEFAULT = 0

If this parameter is set to one, the computer will not print the final tables describing the communication structure of each group. (These tables are described below.)

P21 -- MAX OUTPUT DEFAULT = 15

This parameter is used for debugging the program. Values lower than 15 cause great volumes of cryptic information to be printed. (This parameter will soon become inoperative.)

P32 -- ISOSUP DEFAULT = 1

If this parameter is left at the default value of one, isolates type one will not appear on the link list at the beginning. (These are the nodes that have no links.) To prevent the suppression of these links, set P32 to zero.

P33 -- DETAILS DEFAULT = 0

If P33 is set to one, the computer will print more details about testing both the individuals and the groups for the criteria. This is useful for verifying the "goodness" of the final solution. However, in most cases this information is not needed. As the program is tested more thoroughly, we can be more confident in its results, and this information is less and less valuable.

P35 -- PUNCH DECK DEFAULT = 0

If P35 is set to one, the computer will punch a deck which contains essentially the same information that appears on the very last table. This includes the node number, the role assigned to that node, the group number if the node is a group member, or an isolate or tree node attached to a group member, the integrativeness score for the node, and the "name" for the node. The format is as follows: one card/card image per node; (4I5,2A10). This information appears on the file called "PUNCH."

C. Output

The output from the final section of the program comes in three parts. First, there is a set of tables for each group. Second is a complete listing of all nodes that are not group members. Finally there is a numerically ordered summary of all the nodes, with a description of each node's role and integrativeness score.

We start with the group tables, which are the first to appear. The tables for the first valid group begin with the phrase "START OF INFORMATION FOR GROUP X," where X is the group's identification number. The computer then prints information about each node in the group. The

information for a node begins with a link that says "MEMBER NUMBER X ID = _____" where the node's subject number is X. The node's "name" appears after the "ID=". Following this line is a list of all the links with that node. Each link is described in a line that begins with either "LINK WITH", "LINK TO", or "LINK FROM", followed by the number of the node at the other end of the link. "LINK WITH" means the link is reciprocated. "LINK TO" means that the link is unreciprocated, not being returned by the node at the other end of the link. "LINK FROM" means that the link was not reciprocated, not being returned by the group member whose links are being listed. A "TO" link would be given a "U" in the big link list printed at the beginning of the printout. A :FROM' would be an "A" and a "WITH" would be an "R".

Following the number of the other node is a description of the type of link. This could be either "WITHIN-GRP", which means that the link is to another member of the same group; a "BRIDGE" link, which means it is a link to a member of another group; a "LIAISON" link, which means it is a link to a liaison; or an "OTHER" link, which is a link to a type "other."

To the right of the link type indicator are the letters "WF=" followed by the within-group weighting factor discussed on pages 25 and 26. (This weighting factor equals the number of two-step links connecting the nodes, plus one.) The weighting factor is followed by the strength of the link as reported by the group member being analysed. (If P34 was set to one and the link was reciprocated, the strength will be the mean of both halves of the link.) Finally, the "name" of the node to whom the link goes is printed.

At the end of the description of links for the node is the phrase "INTEGRATIVENESS OF NODE X IS Y" where "X" is the node number and "Y" is the integrativeness of that node. Integrativeness is calculated as number of links between the nodes linked to the original node divided by the largest possible number of such links. The value ranges from zero to one.

Directly below the integrativeness score is a "LINK ANALYSIS MATRIX" which is actually two matrices. The one on the left is calculated using numbers of links, while the one on the right is calculated using strengths of links. Each matrix has five columns and six rows. The first column has values for TWO-WAY (reciprocated) links; the second has values for OUTGOING (unreciprocated) links; and the third for INCOMING (added) links. The fourth column has row totals and the fifth row percentages. The first row has numbers for within-group links, the second for between group links, the third for liaison links, and the fourth for other links. The fifth row has column totals, and the sixth, column percentages.

With these tables, it is easy to make statements about the percentage of within-group linkage that is reciprocated, the relative strengths of within-group links and between-group links, and so on.

After all the nodes in the group have been analyzed, there is a "GROUP LINK ANALYSIS MATRIX." The tables at the top of this set of matrices are identical to the link analysis matrices printed for each individual node, with the exception that all links with group members are included, instead of only links with a single node. In addition to the "NUMBER OF LINKS" and "STRENGTHS OF LINKS" tables, there are two other tables -- "AVERAGE WEIGHTING FACTOR" and "AVERAGE STRENGTH." Both

of these tables have columns for two-way, outgoing, incoming, and total; both have rows for within-group, between-group, liaison, other, and total. The four tables in the group link analysis matrix are set off from the rest of the printout by dashed lines drawn across the page.

At the bottom of the group link analysis matrices is a specification of the type of relation chosen (P04) and a calculation of group connectiveness (density). This value is calculated as the number of within-group links divided by the maximum possible. At this point, the computer moves onto the "START OF INFORMATION" for the next group.

After the last set of connectiveness calculations are the lists of nodes that are not group members. This set includes, in order, isolates type one, isolates type two, isolated dyads, tree nodes, "others," and liaisons. For each category there is a list of nodes that fit that category, with an analysis of all the links for that node. This breakdown specifies who each link is with, what kind of node they are, and what that node's "name" is. When appropriate, the integrativeness of the node is printed along with a breakdown of links into two-way, incoming, and outgoing.

After the last liaison is described, there is a small table telling how many nodes of each type there were. Finally, the last table printed as part of the analysis contains a list of all the nodes, in ascending numerical order, with a specification of the role each node is assigned to, the group they are a member of or attached to, and their integrativeness score and "name." Integrativeness scores range from zero to 1000 on this list; these values must be divided by 1000 to correctly locate

the decimal point. Integrativeness scores for tree nodes having negative values should be ignored. This concludes the output of the Network Analysis Program.

PART FOUR

USING THE NETWORK ANALYSIS PROGRAM

This part is divided into two sections. In the first, the actual running of the Negopy Program is discussed. This section includes the specification of parameters, the setting up of input decks, and so on. The second section covers the less mechanical aspects of using the program: error messages and what they mean, how to interpret strange results, "fine tuning" the program, and known bugs in the program.

I. SETTING UP A NETWORK ANALYSIS RUN

In general, any Network Analysis run includes two kinds of information: control cards and data. The data, of course, include all the cards with information describing the links, as well as a "namelist," if there is one. There are two basic types of control cards -- system control cards and Negopy control cards. The system control cards tell the computer what to do -- which program to execute, how much memory is needed, how long the program may run, and so on. The Negopy control cards set the parameters in the program, describe the format of the data (and the namelist), and provide some other information the program needs to execute the run. Since the system cards come first, they will be discussed first.

System Control Cards

The system control cards will look like either "A" or "B" as shown here.

A	B
JOB,_____ ATTACH,A,NEGOPY. A. 7-8-9 (EOF)	JOB,_____ ATTACH,A,NEGOPY. ATTACH,DATA,yourdatafile. A. 7-8-9 (EOF)

The cards in Set A are used when data are on punched cards. The cards in B are used if the data are on a permanent file called "yourdatafile."

The first card is a job card. Here you specify your account number, time limits, memory length, and so on. In general, memory requirements are about $70K_8$ for small networks, 100 to $110K_8$ for medimum ones, and $150K_8$ or more for very large ones. Experimentation is needed to determine the best numbers to use.

The program usually runs in under three minutes. For very large runs, it is a good idea to allow five to ten minutes, however. The number of pages of output varies roughly as the number of nodes. A lower limit should be about 100 pages. Generally, an overestimate is safer than an underestimate, since a low estimate will necessitate re-running the whole program.

The second card fetches the program object deck which is produced by compiling a source deck (FORTRAN) and cataloging the LGO file.

The next card in B attaches the data file and gives it the local name "DATA." The "A" card causes the computer to begin execution of the program, and the 7-8-9 card (end of file) terminates the system control cards.

Negopy Control Cards

The Negopy input cards look like either A, B, C, or D below.

A	B	C	D
Label Card One Label Card Two Parameter Card Parameter Card ⋮ 7-8-9 Format Card for Data Data ⋮ 7-8-9	Label Card One Label Card Two Parameter Card Parameter Card ⋮ 7-8-9 Format Card for Data 7-8-9	Label Card One Label Card Two Parameter Card Parameter Card ⋮ 7-8-9 Format Card for Data Format Card for Name List Namelist ⋮ 7-8-9 Data ⋮ 7-8-9	Label Card One Label Card Two Parameter Card Parameter Card ⋮ 7-8-9 Format Card for Data Format Card for Name List Namelist ⋮ 7-8-9

The cards shown in A are used when the data are on cards and there is no namelist. The cards in C are used if the data are on cards and there is a namelist. The cards in D are used if the data are on a file and there is a namelist. In the rest of this section, the following topics will be covered: label cards, parameter cards, data cards and data format cards, and namelist cards and namelist format cards.

Label Cards

There are always two label cards submitted with each run. The label cards may contain anything the user cares to put there. Whatever is put on the label cards will be printed from time to time on the printout as identifying information. Therefore, it is useful to describe the dataset being used, the date of analysis, and any other identifying information that may be helpful at some later time.

Parameter Cards

Parameter cards are used to change parameters to values other than default values. If no parameter cards are used, all parameters will remain set at their default values.

There may be up to six parameter cards used in any single run. Usually only one or two will be needed, however. After the last parameter card there is always a 7-8-9 card.

Each parameter card may be divided into eight ten-column fields. One parameter can be set on each card, although any number less than eight may be set on any particular parameter card.

Each ten-column field has this format:

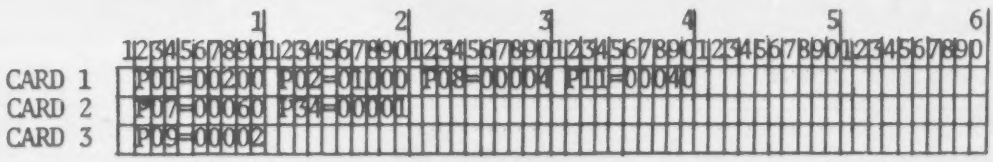
	P	X	X	=	Y	Y	Y	Y	Y
1	2	3	4	5	6	7	8	9	10

The first column (Column 1, or Column 11, 21, 31, etc.) is always empty. The second column (Column 2, 22, 32, 42, etc.) always says "P". The third and fourth columns (Column 3 and 4, 13 and 14, 23 and 24, etc.) have the number of the parameter being set. The fifth column (Column 5, 15, 25, etc.) always has an "=". The remaining five columns in the field have

the value the parameter is to be set to. This value must be right -- justified in the field. For example, if a parameter is being set to a value of one, the value field would either be four blanks followed by a "1" or four zeros and a "1".

The first 35 parameters (P01 to P35) may be set to integer values (i.e., there may or may not be decimal points in the value fields for the first 35 parameters). The last ten parameters (P36 to P45) require decimal points.

As many parameter cards as are needed (up to six) may be used. If eight parameters are to be set, they can all be done on one card. Alternatively, they could be set on two cards, each of which has four parameters. The parameters may be set in any order. No parameter may be set twice. An example of a set of parameter cards is shown below.



In this example, the following parameter settings were made:

P01 = 200	P08 = 4	P07 = 60	P34 = 1
P02 = 1000	P11 = 40	P09 = 2	

Data Cards and Data Format Cards

The format of data cards was discussed earlier on page . A quick review here would be helpful. There can be up to ten links on a data card. The respondent's number must come before any links. Each link requires either two or three numbers to describe it. The first is always

a contactee number. The second is an Xweight, and the third, if it is present, is a Yweight. Both the respondent number and the contactee numbers must be in integer (I) format, while X-and Yweights must be in floating point (F) format. There may be as many blanks as is desired between the various fields. There may be other information before or after any of the numbers on the card, as long as the order of: Respondent #, [Contactee #, Xweight (Yweight)][Contactee #, Xweight (Yweight)] ..... is followed across the card.

For example, Columns 1-5 might have some identifying information like project number. Columns 6-10 might be respondent #. Links could each require ten columns, so that contactee numbers would be in Columns 11-15, 21-25, 31-35, 41-45, 51-55, and 61-65. Xweights might appear in Columns 17-18, 27-28, 37-38, 47-48, 57-58, and 67-68. The other columns are left blank. The format for this coding plan would be:

(5X, I5, 6(I5, X, F2.0, 2X))

The "5X" tells the computer to skip five columns.

The first "I5" tells it that a five-column integer is next (the respondent number).

The "6(" tells the computer that a five-column integer is next (a contactee number).

The next "I5" tells the computer that a five-column integer is next (a contactee number).

The "X" means "skip a column."

The "F2.0" tells the computer that a two-column floating point number, with no places to the right of the decimal, is next (this is an Xweight).

The "2X" means "skip two columns."

The first ")" means "this is the end of the part that is repeated", and the second ")" means "this is the end of the card."

The format is punched on a card, starting in Column One, leaving no spaces between characters. No other information may appear on this card.

Namelist Cards and Namelist Format Cards

If there is a namelist (i.e., P09 is set to "1" or "2"), it will be structured as a list of elements. Each element must have first the node number and second that node's "name." The name may be up to twenty columns wide. Elements may be punched one per card or several per card. The namelist format card will tell the computer how many "names" there are per card.

Example. One "name" per card, with P09 = 1.

(3X, I5, 2X, A10) The computer will get node numbers from Columns 4-8, skip Columns 9 and 10, and get the "names" from Columns 11-20.

Example. Three "names" per card, with P09 = 2.

(3(I5, 2A10)) The "3" tells the computer that there are three elements per card. The first five columns of each element are the node number, and the next twenty columns are the "name."

The namelist is terminated by a 7-8-9 (eof) card. If some nodes do not have "names," the computer will assign blank names to them. The names do not have to be in any particular order within the namelist.

II. MISCELLANEOUS ASPECTS OF RUNNING THE PROGRAM

A. Error Messages or Warnings

The program prints several kinds of error messages and warnings when certain situations are encountered. These will be covered in the order they may appear.

1. "DANGER. YOU ASKED FOR XXX LINKS. YOU ONLY HAD YYY." This message appears after the raw data have been printed. It usually means that an error was made in either the deck set-up (the cards were not in the correct order) or the input format card for the data.

2. "DANGER. YOU HAD MORE LINKS THAN YOU SAID. THE REST WILL BE IGNORED." The meaning of this is self-explanatory. To correct the situation, set P02 to a higher value.

3. "ALL MEANS LESS THAN OR EQUAL TO ZERO. PROGRAM WILL STOP HERE." This indicates wither that the strengths of all links are very low or that there are not enough participants to go on with the analysis. Review the data format to see that the links are being properly read.

4. "THERE ARE NO GROUPS FOR THIS RUN." For one reason or another, there are no groups. This could be due to the data--all nodes are either non-participants or else there is no organization into groups, in which case all nodes will be classified as tyoe "other". The problem could also be due to the way the parameters were set. For example, if the histogram is thin and spotty, the value of P14 should be raised. If the histogram looks like the one shown in "B" on page 62 but there are no horizontal lines drawn across the page, the value of P23 should be raised. Another possibility is that more iterations are needed. P05 should be

set to a higher value -- perhaps eight -- if it was six or lower in the run that gave no groups. If P05 was set to a higher value already, it should be set to a lower value -- perhaps to four.

5. "DANGER. GROUP IS TOO LARGE. GROUP HAS OVER 100 MEMBERS ." The program cannot apply the formal criteria to groups having over 100 members. Such large groups may be evidence of systems with a very low degree of organization. (When random data are submitted to Negopy, monolithic groups result.) If it is felt that this is not the case, P14 can be lowered and P23 raised. If this does not succeed in giving more groups, the links can be "thinned out," perhaps by dropping unreciprocated links or raising the value of P10. If this is not acceptable, the value of P44 may be raised and the run re-submitted.

6. "YOU CERTAINLY HAVE A LOT OF ISOLATES. ARE YOU SURE YOU'RE DOING THIS RIGHT?" This message is printed whenever the number of isolates exceeds a certain percentage of the total. It is just a warning message that indicates that the parameters should be carefully checked for accuracy.

7. "XX PERCENT RECIPROCATION--THAT'S VERY LOW....." This is a warning message similar to the one in 6 above.

B. Adjusting the Parameters to get Better Results

1. If the groups produced are large and loosely linked, with a lot of 5's, 6's, 7's, or even higher numbers in the distance matrix, P43 could be raised and P24 lowered. This will cause the computer to try harder in the splitting of groups.

2. If a large proportion of the nodes are "type other," several things can be done. If P03 is presently set to "1", it could be switched to "0", causing unreciprocated links to be dropped. P10 could be raised to a higher value, causing weak links to be ignored. P44 could be raised to a higher value, weighting links that look like within-group links more in the vector averaging process. P05 could be either raised or lowered.

It may not be possible to eliminate all the "others" if they really do not fit into a group structure configuration. They just might not be organized enough to be differentiated into groups.

3. If the groups are very small, it may be that the computer is splitting them too much. If it is felt that this is the case, the value of P43 should be lowered and the value of P24 raised. If the groups are too small even without splitting by the computer, the value of P23 can be lowered, the value of P14 raised, and the value of P05 raised. If P03 is presently set to "0", it could be changed to "1".

C. Known Bugs

There are still a few minor bugs (errors) in the program. We have tried to locate and fix all of these, but some are especially resistant to fixing. The ones we are aware of are:

1. P12 sometimes prints an error message which says that the value supplied by the user is invalid, even when the user does not supply a value. The program sets P12 to 1 in these cases.

(Note-- this bug has been fixed since the manual was written)

2. Sometimes the computer prints an error message or warning when it is splitting a group into two parts, even when the message does not apply. Ignore the message.

3. When a group exceeds the limit of 100 members, all further information pertaining to that group may be unreliable.

4. The integrativeness score for tree nodes may be negative numbers. Ignore these values. Tree nodes, like all non-participants, have integrativeness scores of zero.

5. Integrativeness scores of members of groups having over 100 members may have "*" in them. Ignore these numbers, as they are unreliable (see Bug # 3).

6. These are all the bugs that are presently known. If you think you have found more, please contact:

WILLIAM D. RICHARDS
Institute for Communication Research
Stanford, California 94305

PHONE: (415) 497-2755

PARAMETER LIST

In this section the control parameters are listed in five parts, which coincide with the five parts of the analysis. In the leftmost column is the parameter number, which is followed by the parameter name and a brief description of the function of the parameter. The numbers in the column labelled "page" indicate which page that parameter is discussed on in the manual. The next column, headed "default" indicates the default value of the parameter--the value that will be supplied by the computer if the user does not set a value. In the last column is the maximum (or minimum) value the parameter can take.

I. DATA DESCRIPTION

			page	default	maximum
P01	# OF NODES	Highest legal subject number	48	0	4095
P02	# OF LINKS	estimate of the number of links	48	0	32,767
P03	RECIPROCAT	0---drop unreciprocated links 1---add links to force reciprocation	49	1	
P04	DIRECTION	0---assume links are non-directed 1---assume links are directed	49	0	
P07	DATA UNIT	40---data are on file with local name "DATA" 60---data are on cards	50	40	
P31	#WEIGHTS	1---only X-weights are used 2---both X-weights and Y-weights	50,51	1	2
P08	#OBSV/CARD	maximum number of links per card	51	1	10
P09	NAME-WIDTH	0---no "names" will be used 1---"names" up to ten columns 2---"names" up to twenty columns	52	0	2

			page	default	maximum
P10	LOW WEIGHT	lowest legal strength for links (after link weighting formula)	52	1	255
P11	HI WEIGHT	highest legal strength for links (after link weighting formula)	52	1	255
P15	EXPONENT	these are all values in the link weighting formula	53	1	4
P37	CONS	"	"	0	
P38	MX	"	"	1	
P39	MY	"	"	0	
P40	CCX	"	"	0	
P41	CCY	"	"	0	
P42	MCPK	"	"	0	
P06	# RAW PRINT	computer prints the first P06 good links	55	10	
P34	MEAN STRST	0---computer uses strength values as reported 1---computer sets both incoming and outgoing strength values for reciprocated links to their mean	55	0	

II. PREPARATION FOR GROUP DETECTION

P44	2-STEP WT	influence of within group weighting factor (see also page 29)	60	1.00
P05	# OF ITERS	number of iterations of vector averaging process	60	4

III. INITIAL GROUP DETECTION

P14	SCAN RADIUS	radius of scanning window	61	200
P13	DNSTY HIST	0---no density histogram printed 1---density histogram will be printed	62	1
P23	GRP-SNSVTY	controls sensivity of group detection routine	62	100

			page	default	maximum
IV. FORMAL GROUP CRITERIA					
P36	PERW	this is the -percentage for groups	65	50.05	
P22	MIN SPLIT	smallest group computer will try to split	65	12	min=5
P24	SPLIT DEV	computer will try to split groups with SD of row means greater than P24	65	30	min=5
P43	DROP-SPLIT	if greater than one, computer will remove up to this many nodes in attempts to split groups. if less than one, computer will remove up to this proportion of group in attempts to split.	66	0.10	
V. FINAL RESULTS AND CONTROL OF OUTPUT					
P17	FILE OUTPUT	0---no file output will be made 1---a file output will be made	68	0	
P18	PRINTO SUP	0---print link list 1---suppress link list	69	0	
P19	GRID SUP	0---print intermediate group lists 1---suppress intermediate lists	69	1	
P20	GROUP SUP	0---print final group tables 1---suppress final group tables	69	0	
P21	MAX OUTPUT	used for debugging the program	69	15	
P32	ISOSUP	0---include isolates in link list 1---suppress isolates in link list	69	1	
P33	DETAILS	0---do not print additional details of group formation process 1---print additional details	70	0	
P35	PUNCH DECK	0---do not punch summary deck 1---punch a summary deck	70	0	